Music Information Retrieval

George Tzanetakis
Associate Professor
Canada Tier II Research Chair
in Computer Analysis of Audio and Music
Department of Computer Science
University of Victoria

January 12, 2017

List of Figures

List of Tables

Contents

Preface

Music Information Retrieval (MIR) is a field that has been rapidly evolving since 2000. It encompases a wide variety of ideas, algorithms, tools, and systems that have been proposed to handle the increasingly large and varied amounts of musical data available digitally. Researchers in this emerging field come from many different backgrounds. These include Computer Science, Electrical Engineering, Library and Information Science, Music, and Psychology. They all share the common vision of designing algorithms and building tools that can help us organize, understand and search large collections of music in digital form. However, they differ in the way they approach the problem as each discipline has its own terminology, existing knowledge, and value system. The resulting complexity, characteristic of interdisciplinary research, is one of the main challenges facing students and researchers entering this new field. At the same time, it is exactly this interdisciplinary complexity that makes MIR such a fascinating topic.

The goal of this book is to provide a comprehensive overview of past and current research in Music Information Retrieval. A considerable challenge has been to make the book accessible, and hopefully interesting, to readers coming from the wide variety of backgrounds present in MIR research. Intuition and high-level understanding are stressed while at the same time providing sufficient technical information to support implementation of the described algorithms. The fundamental ideas and concepts that are needed in order to understand the published literature are also introduced. A thorough bibliography to the relevant literature is provided and can be used to explore in more detail the topics covered in this book. Throughout this book my underlying intention has been to provide enough support material to make any publication published in the proceedings of the conference of the International Society for Music Information Retrieval (ISMIR) understandable by the readers.

I have been lucky to be part of the growing MIR community right from its inception. It is a great group of people, full of passion for their work and always open to discussing their ideas with researchers in other disciplines. If you ever want to see a musicologist explaining Shenkerian analysis to an engineer or an engineer explaining Fourier Transforms to a librarian then try attending ISMIR, the International Conference of the Society for Music Information Retrieval. I hope this book motivates you to do so.

Acknowledgements

Music Information Retrieval is a research topic to which I have devoted most of my adult life. My thinking about it has been influenced by many people whom I have had the fortune to interact with. As is the case in most such acknowledgement sections there are probably more names than most readers care about. At the same time there is a good chance that I also have probably forgotten some to which I apologize.

First of all I would like to thank my two sons Panos and Nikos for not letting my work consume me. I am typing these sentences on an airplane seat with my 1 year old son Nikos sleeping on my lap and my 5 year old son Panos sleeping next to me. Like most kids they showed music information retrieval abilities before they walked or talked. I always smile when I remember Panos bobbing his head in perfect time with the uneven rhythms of Cretan folk music, or Nikos pointing at the iPod until the music started so that he could swirl to his favorite reggae music. Both of these seemingly simple tasks (following a rhythm and recognizing a particular genre of music) are still beyond the reach of computer systems today, and my sons, as well as most children their age, can perform them effortlessly. Since 2000, I and many other researchers have explored how to computationally solve such problems utilizing techniques from digital signal processing, machine learning, and computer science. It is humbling to realize how much further we still need to go.

I would also like to thank my parents Anna and Panos who kindled my love of music and science respectively from an early age. I also would like to thank my uncle Andreas who introduced me to the magic of computer programming using Turbo Pascal (version 3) when I was a high shool student. Studying music has always been part of my life and I owe a lot to my many music teachers over the years. I would like to single out Theodoros Kerkezos, my saxophone teacher, who through his teaching rigor turned a self taught saxophone player

with many bad playing habits into a decent player with solid technique. I have also been fortunate to play with many incredible musicians and I am constantly amazed by what an incredible experience playing music is. I would like to thank, in somewhat chronological order: Yorgos Pervolarakis, Alexis Drakos Ktistakis, Dimitris Zaimakis, Dimitris Xatzakis, Sami Amiris, Tina Pearson and Paul Walde with whom I have shared some of my favorite music making.

I have also been very fortunate to work and learn from amazing academic mentors. As an undergraduate student, my professors Giorgos Tziritas and Apostolos Traganitis at the University of Crete exposed me to research in image and signal processing, and supported my efforts to work on audio and music processing. My PhD supervisor Perry Cook provided all the support someone could wish for, when my research involving computer analysis of audio representations of music signals was considered a rather exotic computer science endeavor. His boundless creativity, enthusiasm, and love of music in most (but not all) forms have always been a source of inspiration. I am also very grateful that he never objected to me taking many music courses at Princeton and learning from amazing music teachers such as Steven Mackey and Kofi Agawu.

I also learned a lot from working as a postdoctoral fellow with Roger Dannenberg at Carnegie Mellon University. Roger was doing music information retrieval long before the term was coined. I am completely in awe of the early work he was able to do when processing sound using computers was so much more challenging than today because of the hardware limitations of the time. When I joined Google Research as visiting faculty during a six month sabbatical leave in 2011, I thought I had a pretty decent grasp of digital signal processing. Dick Lyon showed me how much more I needed to learn especially related to computer models of the human auditory system. I also learned a lot from Ken Steiglitz both as a graduate student at Princeton University and later on through his book “A Digital Signal Processing Primer”. I consider this book the best DSP book for beginners that I have read. It also strongly influenced the way I approach explaining DSP, especially for readers with no previous exposure to it both in the classes that I teach as well as in this book.

Marsyas (**M**usic **A**nalysis, **R**etrieval and **S**ynthesis for **A**udio **S**ignals) is a free software framework for audio processing with specific emphasis on music information retrieval that I designed and for which I am the main developer since 2000. It has provided me, my students and collaborators from around the world, a set of tools and algorithms to explore music information retrieval. The core *Marsyas* developers have been amazing friends and collaborators throughout the years. I would like to particularly acknowledge, in very rough chronological

order: Ari Lazier, Luis Gustavo Martins, Luis F. Texeira, Aaron Hechmer, Neil Burroughs, Carlos Castillio, Mathieu Lagrange, Emiru Tsunoo, Alexander Lerch, Jakob Leben and last but by no means least, Graham Percival all of which made significant contributions to the development of *Marsyas*.

I have also been taught a lot from the large number of undergraduate and graduate students I have worked with with over the course of my academic career. I would like to single out Ajay Kapur and Adam Tindale who through their PhD work made me realize the potential of utilizing music information retrieval techniques in the context of live music performance. My current PhD students Steven Ness and Shawn Trial helped expand the use of automatic audio analysis to bioacoustic signals and robotic music instruments respectively. Graham Percival has been a steady collaborator and good friend through his Masters under my supervision, his PhD at the University of Glasgow, and a 6 month post-doc working with me. I have also had the great fortune to work with several Postdoctoral fellows and international visiting students. Mathieu Lagrange and Luis Gustavo Martins have been amazing friends and collaborators. I also had a great time working with and learning from Alexander Lerch, Jayme Barbedo, Tiago Tavares, Helene Papadopoulos, Dan Godlovich, Isabel Barbancho, and Lorenzo Tardon.

Chapter 1

Introduction

I know that the twelve notes in each octave and the variety of rhythm offer me opportunities that all of human genius will never exhaust.
Igor Stravinsky

In the first decade of the 21st century there has been a dramatic shift in how music is produced, distributed and consumed. This is a digital age where computers are ubiquitous in every aspect of our lives. Music creation, distribution, and listening have gradually become to a large extent digital. For thousands of years music only existed while it was created at a particular time and place. Suddenly not only it can be captured through recordings but enormous amounts of it are easily accessible through electronic devices. A combination of advances in digital storage capacity, audio compression, and increased network bandwidth has made this possible. Portable music players, computers, and smart phones have facilitated the creation of personal music collections consisting of thousands of music tracks. Millions of music tracks are accessible through streaming over the Internet in digital music stores and personalized radio stations. It is very likely that in the near future all of recorded music will be available and accessible digitally.

The central challenge that research in the emerging field of music information retrieval (MIR) tries to address is how to organize and analyze this vast amount of music and music-related information available digitally. In addition, it also involves the development of effective tools for listeners, musicians and musi-

cologists which are informed by the results of this computer supported analysis and organization. MIR is an inherently interdisciplinary field combining ideas from many fields including Computer Science, Electrical Engineering, Library and Information Science, Music, and Psychology. The field has a history of approximately 15 years so it is a relatively new research area. Enabling anyone with access to a computer and the Internet to listen to essentially most of recorded music in human history is a remarkable technological achievement that would have probably been impossible even as recently as 20 years ago. The research area of Music Information Retrieval (MIR) gradually emerged during this time period in order to address the challenge of effectively accessing and interacting with these vast digital collections of music and associated information such as meta-data, reviews, blogs, rankings, and usage/download patterns.

This book attempts to provide a comprehensive overview of MIR that is relatively self-contained in terms of providing the necessary interdisciplinary background required to understand the majority of work in the field. The guiding principle when writing this book was that there should be enough background material in it, to support understanding any paper that has been published in the main conference of the field: the International Conference of the Society for Music Information Retrieval (ISMIR). In this chapter the main concepts and ideas behind MIR as well as how the field evolved historically are introduced.

1.1 Context and History

Music is one of the most fascinating activities that we engage as a species as well as one of the most strange and intriguing. Every culture throughout history has been involved with the creation of organized collections of air pressure waves that somehow profoundly affect us. To produce these waves we have invented and perfected a large number of sound producing objects using a variety of materials including wood, metal, guts, hair, and skin. With the assistance of these artifacts, which we call musical instruments, musicians use a variety of ways (hiting, scraping, bowing, blowing) to produce sound. Music instruments are played using every conceivable part of our bodies and controlling them is one of the most complex and sophicated forms of interaction between a human and an artifact. For most of us listening to music is part of our daily lives, and we find it perfectly natural that musicians spend large amounts of their time learning their craft, or that we frequently pay money to hear music being performed. Somehow

these organized structures of air pressure waves are important enough to warrant our time and resources and as far as we can tell this has been a recurring pattern throughout human history and across all human cultures around the world. It is rather perplexing how much time and energy we are willing to put into music making and listening and there is vigorous debate about what makes music so important. What is clear is that music was, is, and probably will be an essential part of our lives.

Technology has always played a vital role in music creation, distribution, and listening. Throughout history the making of musical instruments has followed the technological trends of the day. For example instrument making has taken advantage of progressively better tools for wood and bone carving, the discovery of new metal alloys, and materials imported from far away places. In addition, more conceptual technological developments such as the use of music notation to preserve and transmit music, or the invention of recording technology have also had profound implications on how music has been created and distributed. In this section, a brief overview of the history of how music and music-related information has been transmitted and preserved across time and space is provided. This overview will help us prepare the ground for understanding modern MIR.

Unlike painting or sculpture, music is emphemeral and exists only during the course of a performance and as a memory in the minds of the listeners. It is important to remember that this has been the case throughout most of human history. Only very recently with the invention of recording technology could music be "preserved" across time and space. The modern term "live" music applied to all music before recording was invented. Music is a process not an artifact. This fleeting nature made music transmission across space and time much more difficult than other arts. For example, although we know a lot about what instruments and to some extent what music scales were used in antiquity, we have very vague ideas of how the music actually sounded. In contrast, we can experience ancient sculptures or paintings in a very similar way to how they were perceived when they were created. For most of human history the only means of transmission and preservation of musical information was direct contact between a performer and a listener during a live performance. The first major technological development that changed this state of affairs was the gradual invention of musical notation. Although several forms of music notation exist for our discussion we will focus on the evolution of Western common music notation which today is the most widely used music notation system around the world.

Several music cultures around the world developed symbolic notation systems although musicians today are mostly familiar with Western common music no-

tation. The origins of western music notation lie in the use of written symbols as mnemonic aids in assisting singers render melodically a particular religious text. Fundamentally the basic idea behind music notation systems is to encode symbolically information about how a particular piece of music can be performed. In the same way that a recipe is a set of instructions for making a particular dish, a notated piece of music is a set of instructions that musicians must follow to render a particular piece of music. Gradually music notation became more precise and able to capture more information about what is being performed. However it can only capture some aspects of a music performance which leads to considerable diversity in how musicians interpret a particular notated piece of music. The limited nature of music notation is because of its discrete symbolic nature and the limited number of possible symbols one can place on a piece of paper and realistically read. Although initially simply a mnemonic aid, music notation evolved and was used as a way to preserve and transmit musical information across space and time. It provided an alternative way of encountering new music to attending a live performance. For example, a composer like J. S. Bach could hear music composed in Italy by simply reading a piece of paper and playing it on a keyboard rather than having to physically be present when and where it was performed. Thanks to written music scores any keyboard player today can still perform the pieces J. S. Bach composed more than 300 hundred years ago by simply reading the scores he wrote. One interesting sidenote that is a recurring theme in the history of technology and music technology is that frequently inventions are used in ways that their original creators never envisioned. Although originally conceived more as a mnemonic aid for existing music, music notation has profoundly affected the way music was created, distributed and consumed. For example, the advent of music notation to a large extent was the catalyst that enabled the creation of a new type of specialized worker which today we know as a composer. Freelance composers supported themselves by selling scores rather than requiring a rich patron to support them and Mozart is a well known example of a composer who experienced the transition from having a single patron to freelancing.

The invention of recording technology caused even more significant changes in the way music was created, distributed and consumed [95, 55]. The history of recording and the effects it had in music is extremely fascinating. In many ways it is also quite informative about many of the debates and discussions about the role of digital music technology today. Recording enabled repeatability of music listening at a much more precise level than music notation ever could and decoupled the process of listening from the process of music making.

Given the enormous impact of recording technology to music it is hard to imagine that there was a time when it was considered a distraction from the "real" application of recording speech for business applications by none less than the distinguished Thomas Edison. Despite his attempts music recordings became extremely popular and jukebox sales exploded. For the first time in history music could be captured, replicated and distributed across time and space and recording devices quickly spread across the world. The ensuing cross-polination of diverse musical styles triggered the generation of various music styles that essentially form the foundation of many of the popular music genres today.

Initially the intention of recordings was to capture as accurately as possible the sound of live music performances. At the same time, limitations of early recording technology such as the difficulty of capturing dynamics and the short duration of recordings forced musicians to modify and adapt how they played music. As technology progressed with the development of inventions such as electric microphones and magnetic tapes, recordings progressively became more and more a different art form. Using effects such as overdubbing, splicing, and reverb, recordings became essentially virtual creations impossible to recreate in live music performance [80].

Early gramophones allowed their owners to make their own recordings but soon after with the standarization and mass production of recording discs the playback devices were limited only to playing. Recordings became something to be collected, prized and owned. They also created a different way of listening that could be conducted in isolation. It is hard for us to imagine how strange this isolated listening to music seemed in the early days of recording. My favorite such story, recounted by Katz [55], is about an early british enthusiast who constructed elaborate dollhouses as visual backdrops for listening to recordings of his favorite operas.

Music broadcasting through radio was another important music technology milestone of the 20th century. It shifted the business model from selling physical recordings to essentially free access to music supported by advertising. It also had an important accessibility effect making music listening effectively free. Swing music, the only jazz genre that became mainstream popular, was to a large extent promoted through radio and made accessible during and after the Great Depression when purchasing individual recordings was prohibitive for many listeners.

Analog casettes reintroduced the lost ability that early grammophones had to perform home recordings. The creation of mix tapes from radio and LP sources was a pervasive and intense activity during the highschool days of the author in the 90s. One can trace the origin of the concept of creating a playlist to mix

tapes. Another way that casettes had impact was the introduction of portable music playing devices starting with the famous Sony Walkman (also one of the prized pocession of the author during high-school in the 90s).

These two historical developments in music technology: musical notation and audio recording, are particularly relevant to the field of MIR, which is the focus of this book. They are directly reflected in the two major types of digital representations that have been used for MIR: symbolic and audio-based. These digital representations were originally engineered to simply store digitally the corresponding analog (for example vinyl) and physical (for example typeset paper) information but eventually were utilized to develop completely new ways of interacting with music using computers. It is also fascinating how the old debates contrasting the individual model of purchasing recordings with the broadcasting model of radio supported by advertising are being replayed in the digital domain with digital music stores that sell individual tracks such as *iTunes* and subscription services that provided unlimited access to streaming tracks such as *Spotify*.

1.2 Music Information Retrieval

As we discussed in the previous section the ideas behind music information and retrieval are not new although the term itself is. In this book we focus on the modern use of the term music information retrieval which will be abbreviated to MIR for the remainder of the book. MIR deals with the analysis and retrieval of music in digital form. It reflects the tremendous recent growth of music-related data digitally available and the consequent need to search within it to retrieve music and musical information efficiently and effectively.

Arguably the birth of Music Information Retrieval (MIR) as a separate research area happened in Plymouth, Massachusetts in 2000 at the first conference (then symposium) on Music Information Retrieval (ISMIR). Although work in MIR had appeared before that time in various venues, it wasn't identified as such and the results were scattered in different communities and publication venues. During that symposium a group of computer scientists, electrical engineers, information scientists, musicologists, psychologists, and librarians met for the first time in a common space, exchanged ideas and found out what types of MIR research were happening at the time. MIR research until then was published in completely separate and different conferences and communities. The seed funding for the symposium and some prepatory workshops that preceded it, was

provided by the National Science Foundation (NSF) through a grant program targeting Digital Libraries (more details can be found at `http://www.ismir.net/texts/Byrd02.html` (accessed January 2014) . The original organizers Don Byrd, J. Stephen Downie and Tim Crawford had backgrounds connected to symbolic processing and digital libraries but fortunately the symposium attracted some attendees that were working with audio. The resulting cross-fertilization was instrumental in establishing MIR as a research area with a strong sense of interdisciplinarity.

When MIR practitioners convey what is Music Information Retrieval to a more general audience they frequently use two metaphors that capture succintly how the majority, but not all, of MIR research could potentially be used. The first metaphor is the so called "grand celestial jukebox in the sky" and refers to the potential availability of all recorded music to any computer user. The goal of MIR research is to develop better ways of browsing, choosing, listening and interacting with this vast universe of music and associated information. To a large extent online music stores, such as the iTunes store, are well on their way to providing access to large collections of music from around the world. A complimentary metaphor could be termed the "absolute personalized radio" in which the listening habits and preferences of a particular user are analyzed or directly specified and computer algorithms provide a personalized sequence of music pieces that is customized to the particular user. As was the case with the traditional jukebox and radio for the first metaphor the emphasis is on choice by the user versus choice by some expert entity for the second metaphor. It is important to note that there is also MIR research that does not fit well in these two metaphors. Examples include: work on symbolic melodic pattern segmentation or work on computational ethnomusicology.

1.3 Dimensions of MIR research

There are multiple different ways that MIR research can be organized into thematic areas. Three particular "dimensions" of organization, that I have found useful when teaching and thinking about the topic are: 1) *specificity* which refers to the semantic focus of the task, 2) *data sources* which refers to the various representations (symbolic, meta-data, context, lyrics) utilized to achieve a particular task, and 3) *stages* which refers to the common processing stages in a

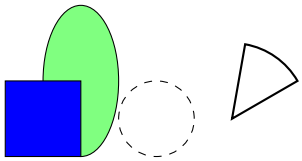


Figure 1.1: Conceptual dimensions of MIR

MIR systems such as representation, analysis, and interaction. In the next three subsections, these different conceptual dimensions are described in more detail.

Make figure here showing the conceptual dimensions of MIR

1.3.1 Specificity

One can view many MIR tasks and systems though a retrieval paradigm in which a query is somehow specified by the user and a set of hopefully relevant results to that query is returned by the computer. This paradigm is very familiar from text search engines such as Google. MIR queries can take many forms such as an actual audio recording, a set of keywords describing the music, or a recording of a user singing a melody. Specificity refers to the range of retrieved results that are considered relevant for a particular query.

The most specific type of task is *audio fingerprinting* in which the query is a recorded (possibly noisy or compressed) audio snippet of a particular music track and the result returned is the exact audio recording from which that snippet was obtained. A different recording of the same piece of music even by the same group is not considered relevant. *Cover song detection* expands the specifity so that the returned results also include different versions of the same piece of music. These can differ in terms of instruments played, style, artist but as long as they have the same underlying melody and harmonic structure they are considered the same.

Further expanding the circle of specifity we have *artist identification* in which any recording by the same artist is considered relevant to a particular query. *Genre* or *style classification* is quite broad and any returned recording that is of the same genre or style is considered relevant. In *Emotion recognition* returned results should all have a particular emotion or evoke a particular mood possibly spanning multiple genres and styles. The most wide circle of specificity is exemplified by simply shuffling in which any music recording in a database is considered

relevant to a particular query. Although shuffling is trivial it is a common way of interacting with large audio collections.

1.3.2 Data sources and information

Another way to conceptually organize MIR tasks and systems is through the data sources and information they utilize. Obviously an important source of information is the music itself or what is frequently termed the musical content. This can be the digital audio recording or some symbolic representation of the music which in most cases can be viewed as a type of a musical score. In addition there is a lot of additional information that can be utilized for retrieval and analysis purposes that is not part of the actual music. This information that is about the music rather than the actual music is termed the context. Frequently it consists of text for example lyrics, blogs, web pages, and social media posts. Other forms of context information can also be used such as ratings, download patterns in peer to peer networks, and graphs of social network connections. Context information tends to be symbolic in nature. Music information retrieval systems can combine multiple source of information. For example, a genre classification system might rely on both audio feature extraction from the recording, and text feature extraction from the lyrics.

1.3.3 Stages

Another way of organizing MIR algorithms and systems is more procedural and through stages that are involved in most such systems. Parallels to these stages can be found in how humans perceive, understand, and interact with music.

- **Representation - Hearing**

  Audio signals are stored (in their basic uncompressed form) as a time series of numbers corresponding to the amplitude of the signal over time. Although this representation is adequate for transmission and reproduction of arbitrary waveforms, it is not particularly useful for analyzing and understanding audio signals. The way we perceive and understand audio signals as humans is based on and constrained by our auditory system. It is well known that the early stages of the human auditory system (HAS), to a first approximation, decompose incoming sound waves into different frequency

bands. In a similar fashion, in MIR, time-frequency analysis techniques are frequently used for representing audio signals. The representation stage of the MIR pipeline refers to any algorithms and systems that take as input simple time-domain audio signals and create a more compact, information-bearing representations. The most relevant academic research area to this stage is Digital Signal Processing (DSP).

- **Analysis - Understanding**

  Once a good representation is extracted from the audio signal various types of automatic analysis can be performed. These include similarity retrieval, classification, clustering, segmentation, and thumbnailing. These higher-level types of analysis typically involve aspects of memory and learning both for humans and machines. Therefore Machine Learning algorithms and techniques are important for this stage.

- **Interaction - Acting**

  When the signal is represented and analyzed by the system, the user must be presented with the necessary information and act according to it. The algorithms and systems of this stage are influence by ideas from Human Computer Interaction and deal with how information about audio signals and collections can be presented to the user and what types of controls for handling this information are provided.

1.4 History

In this section, the history of MIR as a field is traced chronologically mostly through certain milestones.

1.4.1 Pre-history

1.4.2 History 2000-2005

The first activities in MIR were initated through activities in the digital libraries community [5].

1.4.3 History 2006-today

1.4.4 Current status

1.5 Existing tutorials and literature surveys

As a new emerging interdiscipinary area there are not a lot of comprehensive published overviews of MIR research. However there are some good tutorials as well as overview papers for specific sub-areas. There are also published books that are edited collections of chapters written by different authors also usually focusing on specific aspects of Music Information Retrieval. A well written overview, although somewhat dated, focusing on information retrieval and digital library aspects peculiar to music as well as typologies of users and their information needs was written by Nicola Orio in 2006.

1.6 Structure of this book

Even though MIR is a relatiely young research area, the variety of topics, disciplines and concepts that have been explored makes it a hard topic to cover comprehensively. It also makes a linear exposition harder. One of the issues I struggled with while writing this book was whether to narrow the focus to topics that I was more familiar with (for example only covering MIR for audio signals) or attempt to provide a more complete coverage and inevitably write about topics such as symbolic MIR and optical music recognition (OMR) with which I was less familiar. At the end I decided to attempt to cover all topics as I think they are all important and wanted the book to reflect the diversity of research in this field. I did my best to familiarize myself with the published literature in topics removed from my expertise and received generous advice and help from friends who are familiar with them to help me write the corresponding chapters.

The organization and chapter structure of the book was another issue I struggled with. Organizing a particular body of knowledge is always difficult and this is made even more so in an interdisciplinary field like MIR. An obvious organization would be historical with different topics introduced chronologically based on the order they appeared in the published literature. Another alternative would be based an organization based on MIR tasks and applications. However, there are

several concepts and algorithms that are fundamental to and shared by most MIR applications. After several attempts I arrived at the following organization which although not perfect was the best I could do. The book is divided into three large parts. The first, titled Fundamentals, introduces the main concepts and algorithms that have been utilized to perform various MIR tasks. It can be viewed as a crash course in important fundamental topics such as digital signal processing, machine learning and music theory through the lens of MIR research. Readers with more experience in a particular topic can easily skip the corresponding chapter and move to the next part of the book. The second part of the book, titled Tasks, describes various MIR tasks that have been proposed in the literature based on the background knowledge described in the Fundamentals part. The third part, titled Systems and Applications, provides a more implementation oriented exposition of more complete MIR systems that combine MIR tasks as well as information about tools and data sets. Appendices provide supplementary background material as well as some more specific optional topics.

1.6.1 Intended audience

The intended audience of this book is pretty broad encompassing any person interested in music information retrieval research. A minimal set of math requirements would be basic high school math with some probability/statistics, linear algebra and very basic calculus as well. Basic familiarity with computer programming is also assumed. Finally although not strictly required basic familiarity with music theory and notation is also helpful. I have tried to make the exposition as much as possible self-contained and the chapters in the Fundamentals section attempt to build the necessary foundation for good understanding of published MIR literature.

This book has evolved from a set of course notes for CSC475 Music Retrieval Systems, a Computer Science course on MIR taught by the author at the University of Victoria. The students are a combination of 4th year undergraduate students and graduate students mostly in Computer Science but sometimes also in other fields.

1.6.2 Terminology

pitch, intensity, timbre = basic features of musical sound

informatics, retrieval
piece, track, song
database, collection, library
modern popular music, western art music
MIDI
monophonic polyphonic
symbolic vs audio
x(t) vs x[t] lower case of time domain and upper case for frequency domain

1.7 Reproducibility

1.8 Companion website

Chapter 2

Tasks

Like any research area, Music Information Retrieval (MIR) is characterized by the types of problems researchers are working on. In the process of growing as a field, more and hopefully better algorithms for solving each task are constanty designed, described and evaluated in the literature. In addition occassionally new tasks are proposed. The majority of these tasks can easily be described to anyone with a basic understanding of music. Moreover they are tasks that most humans perform effortlessly. A three year old child can easily peform music information related tasks such as recognizing a song, clapping along with music, or listening to the sung words in a complex mixture of instrumental sounds and voice signals. At the same time, the techniques used to perform such tasks automatically are far from simple. It is humbling to realize that a full arsenal of sophisticated signal processing and machine learning algorithms are required to perform these seemingly simple music information tasks using computers. At the same time computers, unlike humans, can process arbitrarily large amounts of data and that way open up new possibilities in how listeners interact with large digital collections of music and associated information. For example, a computer can analyze a digital music collection with thousands of songs to find pieces of music that share a particular melody or retrieve several music tracks that match an automatically calculated tempo. These are tasks beyond the ability of most humans when applied to large music collections. Ultimately MIR research should be all about combining the unique capabilities of humans and machines in the context of music information and interaction.

In this chapter, a brief overview of the main MIR tasks that have been proposed in the literature and covered in this book is provided. The goal is to describe how each task/topic is formulated in terms of what it tries to achieve rather than explaining how these tasks are solved or how algorithms for solving them are evaluated. Moreover, the types of input and output required for each task are discussed. In most tasks there is typically a seminal paper or papers that had a lot of impact in terms of formulating the task and influencing subsequent work. In many cases these seminal papers are preceded by earlier attempts that did not gain traction. For each task described in this chapter I attempt, to the best of my knowledge, to provide pointers to the earliest relevant publication, the seminal publication and some representative examples. Additional citations are provided in the individual chapters in the rest of the book.

This chapter can also be viewed as an attempt to organize the growing and somewhat scattered literature in MIR into meaningful categories and serve as a quick reference guide. The rest of the book is devoted to explaining how we can design algorithms and build systems to solve these tasks. Solving these tasks can sometimes directly result in useful systems, but frequently they form components of larger and more complicated and complex systems. For example beat tracking can be considered a MIR task, but it can also be part of a larger computer-assisted DJ system that performs similarity retrieval and beat alignment. Monophonic pitch extraction is another task that can be part of a larger query-by-humming system. Roughly speaking I consider as a task any type of MIR problem for which there is existing published work and there is a clear definition of what the expected input and desired output should be. The order of presentation of tasks in this chapter is relatively subjective and the descriptions are roughly proportional to the amount of published literature in each particular task.

2.1 Similarity Retrieval, Playlisting and Recommendation

Similarity retrieval (or query-by-example) is one of the most fundamental MIR tasks. It is also one of the first tasks that were explored in the literature. It was originally inspired by ideas from text information retrieval and this early influence is reflected in the naming of the field. Today most people with computers use search engines on a daily basis and are familiar with the basic idea of text

information retrieval. The user submits a query consisting of some words to the search engine and the search engine returns a ranked list of web pages sorted by how relevant they are to the query.

Similarity retrieval can be viewed as an analogous process where instead of the user querying the system by providing text the query consists of an actual piece of music. The system then responds by returning a list of music pieces ranked by their similarity to the query. Typically the input to the system consists of the query music piece (using either a symbolic or audio representation) as well as additional metadata information such as the name of the song, artist, year of release etc. Each returned item typically also contains the same types of meta-data. In addition to the audio content and meta-data other types of user generated information can also be considered such as ranking, purchase history, social relations and tags.

Similarity retrieval can also be viewed as a basic form of playlist generation in which the returned results form a playlist that is "seeded" by the query. However more complex scenarios of playlist generation can be envisioned. For example a start and end seed might be specified or additional constraints such as approximate duration or minimum tempo variation can be specified. Another variation is based on what collection/database is used for retrieval. The term playlisting is more commonly used to describe the scenario where the returned results come from the personal collection of the user, while the term recommendation is more commonly used in the case where the returned results are from a store containing a large universe of music. The purpose of the recommnedation process is to entice the user to purchase more music pieces and expand their collection. Although these three terms (similarity retrieval, music recommendation, automatic playlisting) have somewhat different connotations the underlying methodology for solving them is similar so for the most part we will use them interchangeably. Another related term that is sometimes used in personalized radio in which the idea is to play music that is personalized to the preferences to the user.

One can distinguish three basic approaches to computing music similarity. Content-based similarity is performed by analyzing the actual content to extract the necessary information. Metadata approaches exploit sources of information that are external to the actual content such as relationships between artists, styles, tags or even richer sources of information such as web reviews and lyrics. Usage-based approaches track how users listen and purchase music and utilize this information for calculating similarity. Examples include collaborative filtering in which the commonalities between purchasing histories of different users are exploited, tracking peer-to-peer downloads or radio play of music pieces to evaluate their "hotness" and utilizing user generated rankings and tags.

There are trade-offs involved in all these three approaches and most likely the ideal system would be one that combines all of them intelligently. Usage-based approaches suffer from what has been termed the "cold-start" problem in which new music pieces for which there is no usage information can not be recommended. Metadata approaches suffer from the fact that metadata information is frequently noisy or inaccurate and can sometimes require significant semi-manual effort to clean up. Finally content-based methods are not yet mature enough to extract high-level information about the music.

Evaluation of similarity retrieval is difficult as it is a subjective topic and therefore requires large scale user studies in order to be properly evaluated. In most published cases similarity retrieval systems are evaluated through some related tasks such how many of the returned music tracks belong to the same genre as the query.

2.2 Classification and Clustering

Classification refers to the process of assigning one or more textual labels in order to characterize a piece of music. Humans have an innate drive to group and categorize things including music pieces. In classification tasks the goal is given a piece of music to perform this grouping and categorization automatically. Typically this is achieved by automatically analyzing a collection of music that has been manually annotated with the corresponding classification information. The analysis results are used to "train" models (computer algorithms) that given the analysis results (referred to as audio features) for a new unlabelled music track are able "predict" the classification label with reasonable accuracy. This is referred to as "supervised learning" in the field of machine learning/data mining. At the opposite end of the spectrum is "unsupervised learning" or "clustering" in which the objects of interest (music pieces in our case) are automatically grouped together into "clusters" such that pieces that are similar fall in the same cluster.

There are several interesting variants and extensions of the classic problems of the classification and clustering. In semi-supervised learning the learning algorithm utilizes both labeled data as in standard classification as well as unlabeled data as in clustering. The canonical output of classification algorithms is a single label from a finite known set of classification labels. In multi-label classification each object to be classified is associated with multiple labels both when training and predicting.

When characterizing music several possible such groupings have been used historically as means of organizing music, and computer algorithms that attempt to perform the categorization automatically have been developed. Although the most common input to MIR classification and clustering systems is audio signals there has also been work that utilizes symbolic representations as well as metadata (such as tags and lyrics) and context information (peer-to-peer downloads, purchase history). In the following subsections, several MIR tasks related to classification and clustering are described.

2.2.1 Genres and Styles

Genres or styles are words used to describe groupings of musical pieces that share some characteristics. In the majority of cases these characteristics are related to the musical content but not always (for example in Christian Rap). They are used to physically organize content in record stores and virtually organize music in digital stores. In addition they can be used to convey music preferences and taste as the stereotypical response to the question "What music do you like" which typically goes something like "I like all different kinds of music except X" where X can be classical, country, heavy metal or whatever other genre the particular person is not fond of. Even though there is clear shared meaning among different individual when discussing genres their exact definition and boundaries are fuzzy and subjective. Even though sometimes they are criticized as being driven by the music industry their formation can be an informal and fascinating process [1].

Top level genres like "classical" or "rock" tend to encompass a wide variety of music pieces with the extreme case of "world music" which has enormous diversity and almost no descriptive capability. On the other hand, more specialized genres (sometimes refered to as sub-genres) such Neurofunk or Speed Metal are meaningful to smaller groups of people and in many cases used to differentiate the insiders from the outsiders. Automatic genre classification was one of the earliest classification problems tackled in the music information retrieval literature and is still going strong. The easy availability of ground truth (especially for top-level genres) that can be used for evaluation, and the direct mapping of this task to well established supervised machine learning techniques are the two main reasons for the popularity of automatic genre classification. Genre classification can be cast

[1] `http://www.guardian.co.uk/music/2011/aug/25/origins-of-music-genres-hip-hop` (accessed September 2011)

either as the more common single-label classification problem (in which one out of a set of N genres is selected for unknown tracks) or as a multi-label classification in which a track can belong to more than one category/genre.

[68]

2.2.2 Artist, Group, Composer, Performer and Album Identification

Another obvious grouping that can be used for classification is identifying the artist or group that performs a particular music track. In the case of rock and popular music frequently the performing artists are also the composers and the tracks are typically associated to a group (like the Beatles or Led Zeppelin). On the other hand in classical music there is typically a clearer distinction between composer and performer. The advantage of artist identification compared to genre is that the task is much more narrow and in some ways well-defined. At the same time it has been criticized as identifying the production/recording approach used, rather than the actual musical content (especially in the case of songs from the same album) and also being a somewhat artifical tasks as in most practical scenarios this information is readily available from meta-data.

2.2.3 Mood/Emotion Detection

Music can evoke a wide variety of emotional responses which is one of the reasons it is heavily utilized in movies. Even though culture and education can significantly affect the emotional response of listeners to music there has been work attempting to automatically detect mood and emotion in music. As this is information that listeners frequently used to discuss music and not readily available in most cases there has been considerable interest in performing it automatically. Even though it is occassionally cast as a single-label classification problem it is more appropriately considered a multi-label problem or in some cases a regression in a continuous “emotion” space.

2.2.4 Instrument Recognition, Detection

Monophonic instrument recognition refers to automatically predicting the name/type of a recorded sample of a musical instrument. In the easiest configuration isolated notes of the particular instrument are provided. A more complex scenario is when larger phrases are used, and of course the most difficult problem is identifying the instruments present in a mixture of musical sounds (polyphonic instrument recognition or instrumentation detection). Instrument classification techniques are typically applied to databases of recorded samples.

2.2.5 Tag annotation

The term "tag" refers to any keyword associated to an article, image, video, or piece of music on the web. In the past few years there has been a gradual shift from manual annotation using fixed hierarchical taxonomies to collaborative social tagging where any user can annotate multimedia objects with tags (so called folksonomies) without conforming to a fixed hierarchy and vocabulary. For example, Last.fm is a collaborative social tagging network which collects roughly 2 million tags (such as "saxophone", "mellow", "jazz", "happy") per month [**?**] and uses that information to recommend music to its users. Another source of tags are "games with a purpose" [**?**] where people contribute tags as a by-product of doing a task that they are naturally motivated to perform, such as playing casual web games. For example TagATune [**?**] is a game in which two players are asked to describe a given music clip to each other using tags, and then guess whether the music clips given to them are the same or different.

Tags can help organize, browse, and retrieve items within large multimedia collections. As evidenced by social sharing websites including Flickr, Picasa, Last.fm, and You Tube, tags are an important component of what has been termed as "Web 2.0". The focus of MIR research is systems that automatically predict tags (sometimes called autotags) by analyzing multimedia content without requiring any user annotation. Such systems typically utilize signal processing and supervised machine learning techniques to "train" autotaggers based on analyzing a corpus of manually tagged multimedia objects.

There has been considerable interest in automatic tag annotation in multimedia research. Automatic tags can help provide information about items that have not been tagged yet or are poorly tagged. This avoids the so called "cold-start problem" [**?**] in which an item can not be retrieved until it has been tagged.

Addressing this problem is particularly important for the discovery of new items such as recently released music pieces in a social music recommendation system.

2.2.6 Other

Audio classification techniques have also been applied to a variety of other areas in music information retrieval and more generally audio signal processing. For example they can be used to detect which parts of a song contain vocals (singing identification) or detect the gender of a singer. They can also be applied to the classification/tagging of sound effects for movies and games (such as door knocks, gun shots, etc). Another class of audio signals to which similar analysis and classification techniques can be applied are bioacoustic signals which are the sounds animals use for communication. There is also work on applying classification techniques to symbolic representations of music.

2.3 Rhythm, Pitch and Music Transcription

Music is perceived and represented at multiple levels of abstraction. At the one extreme, an audio recording appears to capture every detailed nuance of music performance for prosperity. However, even in this case this is an illusion. Many details of a performance are lost such as the visual aspects of the performers and musician communication or the intricacies of the spatial reflections of the sound. At the same time, a recording does capture more concrete, precise information than any other representation. At the other extreme a global genre categorization of a music piece such as saying this is a piece of reggae or classical music reduces a large group of different pieces of music that share similar characteristics to a single world. In between these two extremes there is a large number of intermediate levels of abstraction which can be used to represent the underlying musical content.

One way of thinking about these abstractions is as different representations that are invariant to transformations of the underlying music. For example most listeners, with or without formal musical training, can recognize the melody of Twinkle, Twinkle Little Star (or some melody they are familiar with) independently of the instrument it is played on, or how fast it is played, or the starting pitch. Somehow the mental representation of the melody is not affected by these

rather drastic changes to the underlying audio signal. In music theory a common way of describing music is based on rhythm and pitch information. Furthermore, pitch information can be abstracted as melody and harmony. A western common music score is a comprehensive set of notation conventions that represent what pitches should be played, when they should be played and for how long, when a particular piece of music is performed. In the following subsections we describe some of the MIR tasks that can be formulated as automatically extracting a variety representations at the different levels of musical abstraction.

2.3.1 Rhythm

Rhythm refers to the hierarchical repetitive structure of music over time. Although not all music is structured rhythmically, a lot of music in the world is and has patterns of sounds that repeat periodically. These repeating patterns are typically formed on top of a underlying conceptual sequence of semi-regular pulses that are grouped hierarchically.

Automatic rhythm analysis tasks attempt to extract different kinds of rhythmic information from audio signals. Although there is some variability in terminoloy there are certain tasks that are sufficiently well-defined and for which several algorithms have been proposed.

Tempo induction refers to the process of extracting an estimate of the tempo (the frequency of the main metrical level i.e the frequency that most humans would tap their foot when listening to the music) for a track or excerpt. Tempo induction is mostly meaningful for popular music with a relatively steady beat that stays the same throughout the audio that is analyzed. Beat tracking refers to the more involved process of identifying the time locations of the individual beats and is still applicable when there are significant tempo changes over the course of the track being analyzed.

In addition to the two basic tasks of tempo induction and beat tracking there are additional rhythm-related tasks that have been investigated to a smaller extent. Meter estimation refers to the process of automatically detecting the grouping of beat locations into large units called measures such as 4/4 (four beats per measure) or 3/4 (3 beats per measure). A related task is the automatic extraction of rhythmic patterns (repeating sequences of drum sounds) typically found in modern popular music with drums. Finally drum transcription deals with the extraction of a “score” notating which drum sounds are played and when in piece of music. Early work focused on music containing only percussion [**?**] but more

recent work considers arbitrary polyphonic music. There is also a long history of rhythmic analysis work related to these tasks performed on symbolic representations (mostly MIDI) in many cases predating the birth of MIR as a research field.

2.3.2 Melody

Melody is another aspect of music that is abstract and invariant to many transformations. Several tasks related to melody have been proposed in MIR. The result of monophonic pitch extraction is a time-varying estimate of the fundamental frequency of a musical instrument or human voice. The output is frequently referred to as a *pitch contour*. Monophonic pitch transcription refers to the process of converting the continuous-valued pitch contour to discrete notes with a specified start and duration. Figure **??** shows visually these concepts. When the music is polyphonic (more than one sound source/music instrument is present) then the related problem is termed predominant melody extraction. In many styles of music, especially modern pop and rock, there is a clear leading melodic line typically performed by a singer. Electric guitars, saxophones, trumpets also frequently perform leading melodic lines in polyphonic music. Similarly to monophonic melody extraction in some cases a continuous fundamental frequency contour is extracted and in other cases a discrete set of notes with associated onsets and offsets is extracted.

A very important and not at all straightforward task is melodic similarity. Once these melodic representations are computed it investigates how they can be compared with each other so that melodies that would be perceived by humans as associated with the same piece of music or as similar are automatically detected as such. The reason this task is not so simple is that there is an enormous number of variations that can be applied to a melody while it still maintains its identity. These variations range from the straightforward transformations of pitch transposition and changing tempo to complicated aspects such as rhythmic elaboration and the introduction of new pitches.

Make figure of melodic variations

[8]

2.3.3 Chords and Key

Chord Detection, Key Detection/tonality

2.4 Music Transcription and Source Separation

Ultimately music is an organized collection of individual sound sources. There are several MIR tasks that can be performed by characterizing globally and statistically the music signal without individually characterizing each individual sound source. At the same time it is clear that as listeners we are able to focus on individual sound sources and extract information from them. In this section we review different approaches to extracting information for individual sound sources that have been explored in the literature.

Source separation refers to the process of taking as input a polyphonic mixture of individual sound sources and producing as output the individual audio signals corresponding to each individual sound source. The techniques used in this task have their origins in speech separation research which frequently is performed using multiple microphones for input. Blind sound separation refers ... .

blind source separation, computational auditory scene analysis, music understanding

2.5 Query-by-humming

In query by humming (QBH) the input to the system is a recording of a user singing or humming the melody of a song. The output is an audio recording that corresponds to the hummed melody selected from a large database of songs. Earlier systems constrained the user input, for example by requiring the user to sing usign fixed syllables such as La, La, La or even just the rhythm (query by tapping) with the maching being against small music collections. Current systems have removed these restrictions accepting as input normal singing and performing the matching against larger music collections. QBH is particularly suited for smart phones. Some related tasks that have also been explored are query by beat boxing in which the input is a vocal rendering of a particular rhythmic pattern and the output is either a song or a drum loop that corresponds to the vocal rhythmic pattern.

[**?**]

2.6 Symbolic Processing

representations [37]
searching for motifs/themes, polyphonic search

2.7 Segmentation, Structure and Alignment

[**?**] [32]
thumbnailing, summaring

2.8 Watermarking, fingerprinting and cover song detection

2.9 Connecting MIR to other disciplines

The most common hypothetical scenario for MIR systems is the organization and analysis of large music collections of either popular or classical music for the average listener. In addition there has been work in connecting MIR to other disciplines such as digital libraries [5, 26], musicology [13].

2.10 Other topics

Computational musicology and ethnomusicology, performance analysis such as extracting vibrato from monophonic instruments [9], optical music recognition [**?**], digital libraries, standards (musicXML), assistive technologies [36], symbolic musicology, hit song science

Part I

Fundamentals

Chapter 3

Audio representations

Is it not strange that sheep's guts should hale souls out of men's bodies
Shakespeare

The most basic and faithful way to preserve and reproduce a specfic music performance is as an audio recording. Audio recordings in analog media such as vinyl or magnetic tape degrade over time eventually becoming unusable. The ability to digitize sound means that extremely accurate reproductions of sound and music can be retained without any loss of information stored as patterns of bits. Such digital representations can theoretically be transferred indefinitely across different physical media without any loss of information. It is a remarkable technological achievement that all sounds that we hear can be stored as an extremely long string of ones and zeroes. The automatic analysis of music stored as a digital audio signal requires a sophisticated process of distilling information from an extremely large amount of data. For example a three minute song stored as uncompressed digital audio is represented digitally by a sequence almost of 16 million numbers (3 (minutes) * 60 (seconds) * 2 (stereo channels) * 44100 (sampling rate)) or 16 * 16 million = 256 million bits. As an example of extreme distilling of information in the MIR task of tempo induction these 16 million numbers need to somehow be converted to a single numerical estimate of the tempo of the piece.

In this chapter we describe how music and audio signals in general can be represented digitally for storage and reproduction as well as describe audio representations that can be used as the basis for extracting interesting information from these signals. We begin by describing the process of sound generation and our

amazing ability to extract all sorts of interesting information from what we hear. In order for a computer to analyze sound, it must be represented digitally. The most basic digital audio representation is a sequence of quantized pulses in time corresponding to the discretized displacement of air pressure that occured when the music was recorded. Humans (and many other organisms) make sense of their auditory environment by identifying periodic sounds with specific frequencies. At a very fundamental level music consists of sounds (some of which are periodic) that start and stop at different moments in time. Therefore representations of sound that have a separate notion of time and frequency are commonly used as the first step in audio feature extraction and are the main topic of this chapter. We start by introducing the basic idea of a time-frequency representation. The basic concepts of sampling and quantization that are required for storing sound digitally are then introduced. This is followed by a discussion of sinusoidal signals which are in many ways fundamental to understanding how to model sound mathematically. The discrete Fourier transform (DFT) is one of the most fundamental and widely used tools in music information retrieval and more generally digital signal processing. Therefore, its description and interpretation forms a large part of this chapter. Some of the limitations of the DFT are then discussed in order to motivate the description of other audio representations such as wavelets and perceptually informed filterbanks that conclude this chapter.

3.1 Sound production and perception

Sound and therefore music is created by moving objects whose motion results in changes in the surrounding air pressure. The changes in air pressure propagate through space and if they reach a flexible object like our eardrums or the membrane of a microphone they cause that flexible object to move in a similar way that the original sound source did. This movement happens after a short time delay due to the time it takes for the "signal" to propagate through the medium (typically air). Therefore we can think of sound as either time-varying air-pressure waves or time-varying displacement of some membrane like the surface of a drum or the diaphragm of a microphone. Audio recordings capture and preserve this time-varying displament information and enable reproduction of the sound by recreating a similar time-varying displament pattern by means of a loudspeaker with an electrically controlled vibrating membrane. The necessary technology only became available in the past century and caused a major paradigm shift

in how music has been created, distributed and consumed [**?**, **?**]. Depending on the characteristics of the capturing and reproduction system a high-fidelity reproduction of the original sound is possible.

Extracting basic information about music from audio recordings is relatively easy for humans but extremely hard for automatic systems. Most non-musically trained humans are able, just by listing to an audio recording, to determine a variety of interesting pieces of information about it such the tempo of the piece, whether there is a singing voice, whether the singer is male or female, and what is the broad style/genre (for example classical or pop) of the piece. Understanding the words of the lyrics despite all the other interferring sounds from the music instruments playing at the same time, is also impressive and beyond the capabilities of any automatic system today. Building automatic systems to perform these seemingly easy tasks turns out to be quite challenging and involves some of the state-of-the-art algorithms in digital signal processing and machine learning. The goal of this chapter is to provide an informal, and hopefully friendly, introduction to the underlying digital signal processing concepts and mathematical tools of how sound is represented and analyzed digitally.

The auditory system of humans and animals has evolved to interpret and understand our environment. Auditory Scene Analysis is the process by which the auditory system builds mental descriptions of complex auditory environments by analyzing sounds. A wonderful book about this topic has been written by retired McGill psychologist Albert Bregman [14] and several attempts to computationally model the process have been made [**?**]. Fundamentally the main task of auditory perception is to determine the identity and nature of the various sound sources contributing to a complex mixture of sounds. One important information cue is that when nearly identical patterns of vibration are repeated many times they are very likely to originate from the same sound source. This is true both for macroscopic time scales (a giraffe walking, a human paddling) and microscopic time scales (a bird call, the voice of a friend). In the microscopic time scales this perceptual cue becomes so strong that rather than perceiving individual repeating sounds as seperate entities we fuse them into one coherent sound source giving rise to the phenomenon of pitch perception. Sounds such as those produced by most musical instruments and the human voice are perceived by our brains as coherent sound producing objects with specific characteristics and identity rather than many isolated repeating vibrations in sound pressure. This is by no means a trivial process as researchers who try to model this ability to analyze complex auditory scenes computationally have discovered [**?**].

In order to analyze music stored as a recorded audio signal we need to devise representations that roughly correspond to how we perceive sound through the auditory system. At a fundamental level such audio representations help determine when things happen (*time*) and how fast they repeat (*frequency*). Therefore the foundation of any audio analysis alogirthm is a representation that is organized around the "dimensions" of *time* and *frequency*.

3.2 Sampling, quantization and the time-domain

The changes in displacement over time that correspond to sound waves are continuous. Originally recordings stored these changes on an analog medium (for example by having a needle engrave a groove) preserving their continuous nature. In order for computers to process audio signals these continuous signals have to be converted into sequences of numbers that are evenly spaced in time and are discrete. There are two steps in that process: sampling and quantization. Sampling corresponds to measuring the continuous values of the signal at discrete instances of time. For example one can "sample" the temperature during the course of a day by recording the value of the temperature every hour using a thermometer. We would then have 24 measurements per day. When sampling sound waves using a computer typically 44100 "measurements" are taken every second. These continuous measurements need to be turned into a discrete representation suitable for digital storage and transmission. This process is called quantization and it is typical to use 16 bits (i.e $2^{16} = 65536$ possible levels) to represent each measurement. The sampling process is characterized by the sampling rate measured in Hertz (for example 44100 Hz) and the bit depth (the number of bits used to represent each measurement for example 16 bit).

The process of sampling and quantization is far from trivial but explaining it is beyond the scope of this book. A fascinating result is that the original continuous signal can be perfectly reconstructed from a corresponding sequence of digital samples provided that the sampling rate is high enough for the signal of interest. In modern computers the process of sampling and quantization and the inverse process of generating a continuous signal is performed with hardware circuits called analog-to-digital converters (ADC) and digital-to-analog converters (DAC). For our purposes the audio signal is be represented as a very long sequence of floating point numbers (between -1 and 1) that we can notate as follows:

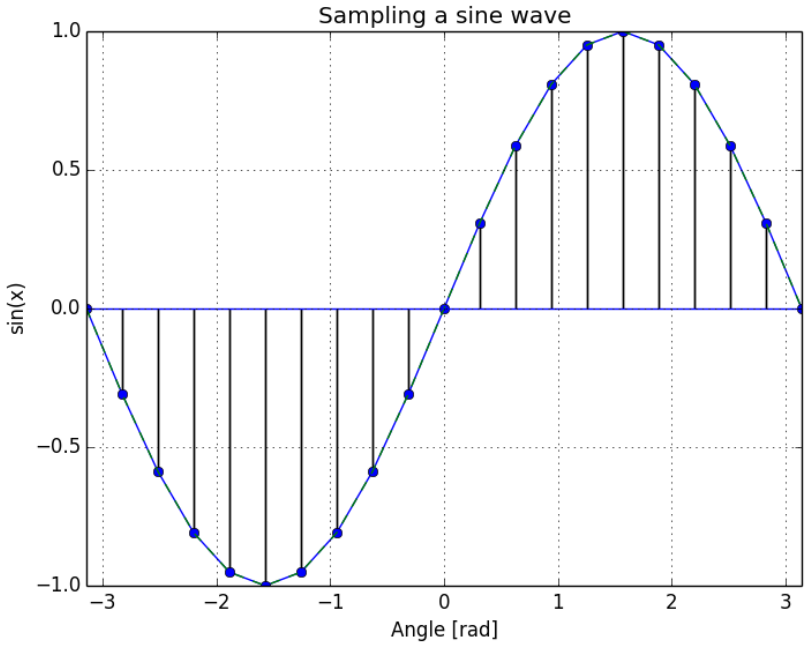


Figure 3.1: Sampling and quantization

$$x[n] \quad n = 0, \ldots, N-1 \tag{3.1}$$

Figure 3.1 shows a continuous sine wave signal in blue and the corresponding sampled values as vertical lines. The height of each line has to be encoded digitally so a discrete set of quantization steps needs to be utilized to represent each value.

3.3 Sinusoids and frequency

Sinusoidal signals are one of the most fundamental abstractions required for understanding how to represent sound digitally as a long series of numbers as well as a way to model mathematically the concept of frequency. In addition they are also fundamental in understanding the mathematical concepts and notation needed for

analyzing and manipulating audio signals so we start our exposition by discussing sinusoids.

Most readers of this book probably have a rough intuitive understanding of a spectrum as a way of measuring/showing the amount of different frequencies present in a signal. For example we know what is the effect of using a graphic equalizer to boost low frequencies. At the same time if someone is not familiar with digital signal processing then equations such as the definition of the Fourier transform:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \tag{3.2}$$

are intimidating and hard to interpret. Frequently even students that have taken courses on the topic end up manipulating the symbols mechanically without a deeper understanding of the underlying concepts. In the following subsections we will attempt to explain the fundamental ideas underlying spectral analysis and the notation used to expresss them.

3.3.1 Sinusoids and phasors

Sinusoidal signals are functions of a particular shape that have some rather remarkable properties. For example one of the most fundamental ideas in digital signal processing is that any sound we hear can be represented as a summation of elementary sinusoidal signals. We motivate sinusoidal signals through several different complimentary ways of viewing them:

- As solutions to the differential and partial differential equations describing the physics of simple vibrating sound generating systems and as good approximation of more complex vibrating systems.

- As the only family of signals that pass in some way unchanged through linear time-invariant systems (LTI). More specifically passing a sine wave of a particular frequency through a LTI system will result in a sine wave of the same frequency possibly scaled and shifted in phase i.e the basic "shape" of the signal remains the same. Many of the systems involved in the production, propagation and sensing of sound can be modeled with reasonable accuracy as linear time-invariant systems and understanding their response to sine waves can be very informative about what these systems do.

- As phasors (rotating vectors) to form an expressive notation that helps intuition and understanding of many fundamental digital signal processing concepts.

- As the basis functions of the Fourier Transform which can be used to represent any periodic signal as a summation of elementary sinusoidal signals.

The term sinusoid is used to describe a family of elementary signals that have a particular shape/pattern of repetition. The sine wave $x(t) = sin(\omega t)$ and the cosine wave $x(t) = cosin(\omega t)$ are particular examples of sinusoids that can be described by the more general equation:

$$x(t) = sin(\omega t + \phi) \tag{3.3}$$

where ω is the frequency and ϕ is the phase. There is an infite number of continuous periodic signals that belong to the sinusoid family but essentially they all have the same shape. Each particular member is characterized by three numbers: the amplitude which encodes the maximum and minimum value of the oscillation, the frequency which encodes the period of repetition, and the phase which encodes the initial position at time $t = 0$. Figure 3.2 shows two sinusoid signals over a time interval of 3 seconds. The first signal (S1) has an amplitude of 7 in arbitrary units, a frequency of $2Hz$, and phase equal to 0. The second signal (S2) has an amplitude of 4, a frequency of $3Hz$, and a phase equal to $\frac{\pi}{4}$. The two sinusoids are also shown superimposed (S1,S2) and their summation is also plotted (S1+S2).

3.3.2 The physics of a simple vibrating system

Music consists of a mixture of sounds many of which are periodic and are produced by musical instruments. Sound is generated by vibrations so a good place to start is with one of the simplest systems in physics that can vibrate. Let's consider striking a metal rod such as the tine of a tuning fork or hitting a ruler that is clamped with a hand on a table. Ther ruler deforms when it is stuck, then a force restores it briefly to its original shape. However it still has inertia and it overshoots the resting position by deforming in the opposite direction. This oscillatory pattern repeats causing the surrounding air molecules to move and the resulting air pressure waves reach our ears as sound. The vibration at any time is caused by the balance between two factors: the force that tends to restore the shape to equilibrium, and the intertia that tends to make it overshoot. This type

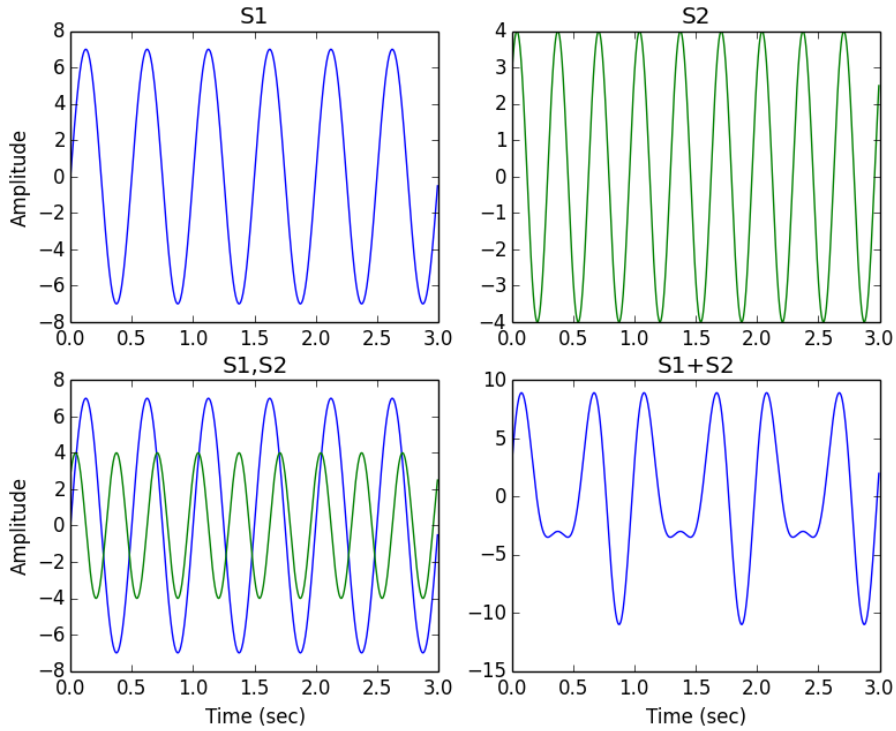


Figure 3.2: Simple sinusoids

of motion is called simple harmonic motion and the equation describing it can be derived from simple physics.

At any time t during the vibration Newton's second law of motion applies: the restoring force produces an acceleration proportional to that force. The restoring force caused by the deformation tries to restore the ruler to its original position, and therefore acts in a direction opposite to the displacement x and for small amounts of deformation its relation to the displacement can be considered linear. This can be written as follows:

$$F = ma = -kx \tag{3.4}$$

Because acceleration is the second derivative with respect to the time variable t we can rewrite this equation as:

$$\frac{d^2x}{dt^2} = -(k/m)x \tag{3.5}$$

where m is the mass, k is the stiffness, x is the displacement from the resting position and t is time. Now we need to figure out a particular function or family of functions that satisfy the equation above. From calculus we know that the derivatives and second derivatives of the sine and cosine functions of t are also sinusoids:

$$\frac{d}{dt}\sin(\omega t) = \omega\cos(\omega t) \qquad \frac{d^2}{dt^2}\sin(\omega t) = -\omega^2\sin(\omega t) \tag{3.6}$$

$$\frac{d}{dt}\cos(\omega t) = -\omega sin(\omega t) \qquad \frac{d^2}{dt^2}\cos(\omega t) = -\omega^2\cos(\omega t) \tag{3.7}$$

where ω is the frequency of oscillation. The only difference between the functions $\sin(\omega t)$ and $\cos(\omega t)$ is that they differ by a delay/time shift but essentially they have the same shape. The term sinusoid will be used when it is not important to distinguish between sine and cosine. As can be seen both of these functions of t satisfy equation 3.5 that characterizes simple harmonic motion. It is also straightforward to show that the equation is satisfied by any sinusoid with an arbitrary phase ϕ of the following form:

$$x(t) = sin(\omega t + \phi) \tag{3.8}$$

Therefore sinusoidal signals arise as solutions to equations that that characterize simple (or simplified) sources of sound vibration such as a tuning fork.

3.3.3 Linear Time Invariant Systems

A general system takes as input one or more input signals and produces as output one or more output signals. We will focus on single input/single output (SISO) systems as they can be used for many operations on audio signals. In addition more complicated MIMO (multiple-input, multiple-output) systems can be assembled by connecting elementary SISO systems together in processing networks. A very important class of SISO systems are the Linear Time-Invariant (LTI) systems because the can be used to model many physical processes and the LTI property makes them easier to handle mathematically. Let's use the following notation: $y(t)$ is the system output, S is the system and $x(t)$ is the

system input. Linearity means that one can calculate the output of the system to the sum of two input signals by summing the system outputs for each input signal individually. Formally if $y_1(t) = S\{x_1(t)\}$ and $y_2(t) = S\{x_2(t)\}$ then $S\{x_1(t)+x_2(t)\} = y_{sum}(t) = y_1(t)+y_2(t)$. Time invariance (or shift invariance in discrete systems) means that the output of the system given an input shifted in time is the same as the output of the system of the original unshifted input signal shifted by the same amount. Formally if $y(t) = S[x(t)]$ then $S[x(t - t_0)] = y(t - t_0)$ where t_0 is the amount of shifting in time. As we will see shortly, frequently we can express complicated signals of interest as linear combinations of more elementary signals. When processing such complicated signals through a LTI system we can analyze its response simply by looking at how it responds to the elementary signals and then combining these individual responses.

Now comes the interesting part which is what happens when a sinusoidal signal is used as input to a LTI system. It turns out that no matter what LTI system is used, the output not only will also be a sinusoidal signal but also have the same frequency as the original input sinusoid. By extension if the input to a LTI system is a linear combination of sinusoidal components of certain frequencies, no new frequencies will be introduced in the output. Therefore we can completely characterize a LTI system by how it responds to elementary sinusoidal signals. Several processes in sound generation and perception can be modelled by LTI systems. Examples include the resonant cavities of musical instruments such as the body of a guitar, parts of the human hearing process such as the the effect of the outer ear, and acoustic spaces such as a concert hall. So far we have motivated sinusoidal signals from physics of simple vibrations and from LTI systems. Now we look into the mathematical notation and concepts that we will need.

3.3.4 Complex Numbers

A good intuitive understanding of sinusoidal signals and the mathematical notation conventions used to represent and manipulate them, is important in order to understand how time-frequency representations work. The notation conventions typically used are based on complex numbers which can intimidate readers without a background in digital signal processing and mathematics. Thinking of sinusoidal signals as vectors rotating at a constant speed on the plane, rather than as single-valued signals that go up and down will help us demystify the complex number notation as well as be able to explain interesting properties geometrically rather than algebraically.

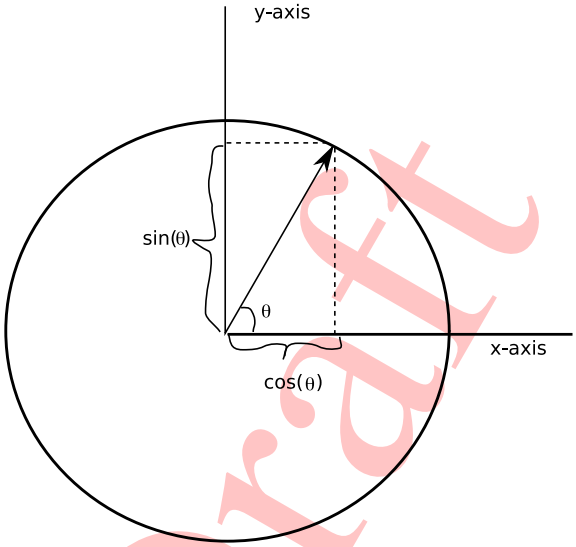


Figure 3.3: Cosine and sine projections of a vector on the unit circle (XXX Figure needs to be redone)

Figure 3.3 shows how the projection of a vector on the unit cycle with angle θ onto the x-axis is the cosine of the angle and the projection onto the y-axis is the sine of the angle. If that vector rotates at constant speed and we plot the x and y axis projections over time the resulting single-valued waveforms are a cosine wave and a sine wave respectively. So we can think of any sine wave as a rotating clock hand or the spoke of bicycle wheel.

Complex numbers provide an elegant notation system for manipulating rotating vectors. Any vector with coordinates x and y is notated as the complex number $x + jy$. The x part is called the *real* part and the part with the weird j factor is called the imaginary part. Typically j is defined as $\sqrt{-1}$. This somewhat strange looking definition makes much more sense if we think of j as meaning rotate by $\frac{\pi}{2}$ (counter-clockwise). Starting from the real number 1 (with 0 imaginary part) two succesive rotations bring us to the real number -1 hence $j^2 = -1$. This should make it clear that there is nothing imaginary or complex about j if it is considered as a rotation.

In terms of addition complex numbers behave exactly as vectors. The x-coordinate of the sum of two vectors is the sum of the x-coordinates and the y-coordinate of the sum of two vectors is the sum of the y-coordinates. Similarly the real part of the sum is the sum of the real parts and the imaginary part of the sum is the sum of the imaginary parts. The expressive power of complex numbers shows up when they are multiplied. We define multiplication using standard algebra and taking care to replace j^2 with -1 whenever we need to. Suppose we want to multiply the complex number $a + jb$ with the complex number $c + jd$. The result is the following complex number:

$$(a + jb) * (c + jd) = ac + jbc + jad + j^2bd = (ac - bd) + j(ad + bc) \quad (3.9)$$

Notice that the use of j is simply a convenient notation so that we can manipulate expressions of complex numbers using our familiar addition and multiplication skills for real numbers. When writing computer code j is not needed and we could write the definition directly as:

```
S.re = A.re * B.re - A.im * B.im

S.im = A.re * B.im + B.re * A.im
```

In order to understand the effect of complex number multiplication we need to think of complex numbers as vectors in *polar* form. The length of the complex

number $c = a + jb$ is called the *magnitude*, is written as $|c|$, and is the Euclidean distance from the origin interpreting a and b as the x and y coordinates:

$$R = |c| = \sqrt{a^2 + b^2} \tag{3.10}$$

The angle with the real axis is often called its *argument* and is given by:

$$\theta = ARG(c) = arctan(b/a) \tag{3.11}$$

Based on these definitions a complex number can be notated as $R\angle\theta$. You can visualize this as scaling the unit vector on the real-axis by R and then rotating the result counter-clockwise by angle $theta$ in order to get to a particular point on the 2D plane. As examples the complex number $1 + j0$ is $1\angle 0$ and the complex number $0 + j1$ is $1\angle\frac{\pi}{2}$. We can easily go back to cartesian coordinates based on the relations shown in Figure 3.3.

$$a = R\cos\theta \qquad b = R\sin\theta \tag{3.12}$$

When complex numbers are represented in polar form their multiplication has a clear geometric interpretation. The magnitudes are multiplied and the angles are added i.e the product of $R_1\angle\theta_1$ and $R_2\angle\theta_2$ is $R_1R_2\angle(\theta_1 + \theta_2)$. Visually multiplication by a compex number is simply a rotation and scaling operation. I hope that after reading this section you can see complex numbers are not as complex if viewed from the right angle.

3.3.5 Phasors

Let's return to our representation of sinusoids as rotating vectors with constant speed and try to use what we have learned about complex numbers for expressing them mathematically. A complex sinusoid can be simply notated as:

$$\cos(\omega t) + j\sin(\omega t) \tag{3.13}$$

where ω is the frequency in radians per second. Every time the time t changes the rotating vector moves around the circle. For example, when the change is $\frac{2\pi}{\omega}$ the argument ωt changes by 2π radians and the sinusoidal signal goes through one cycle.

Another way to view this process is that the rotating vector that represents a sinusoid is just a single fixed complex number raised to progressively higher and

higher powers as time goes by. As we have seen complex number multiplication can be interpreted as a rotation so raising a complex number to an integer power corresponds to moving the phasor around the circle in discrete steps. By making the power continuous and a function of time we can model a continuous rotation. More specifically we can notate our rotating phasor as a function of continuous angle (assuming unit magnitude):

$$E(\theta) = C^{\theta} = \cos\theta + j\sin\theta = 1\angle\theta \tag{3.14}$$

where

$$\theta = \omega t \tag{3.15}$$

We can easily calculate the derivative of this complex function $E(\theta)$ with respect to θ:

$$\frac{dE(\theta)}{d\theta} = -\sin\theta + j\cos(\theta) = jE(\theta) \tag{3.16}$$

$$e^{j\omega t} = \cos\omega t + j\sin(\omega t) \tag{3.17}$$

Hopefully after this exposition the complex number notation used in the definition of various frequency transforms will not look as intimidating. Another possible viewpoint is to simply consider the use of exponentials as a notation convention that allows us to express complex number multiplication using the regular rules we expect from exponents. Using this convention the multiplication of two complex numbers c_1 and c_2 can be written as:

$$c_1 \cdot c_2 = R_1 e^{j\theta_1} R_2 e^{j\theta_2} = R_1 R_2 e^{j(\theta_1+\theta_2)} \tag{3.18}$$

We end this subsection by returning to the geometric view of phasors as vectors rotating at constant speed and using it to illustrate intuitive explanations of some important properties of sinusoidal signals.

XXX geometric phasor descriptions of negative frequency and cancelling of the imaginary part, adding sinusoids of the same frequency but with different amplitude and phase, aliasing and associated figures XXX

3.3.6 Time-Frequency Representations

A large number of organisms including humans are capable of both producing and detecting sound (variations in air pressure). The two most important questions a listener needs to answer regarding any sound are when it happened and what

produced it. One important cue about the origin of a sound is the rate of the periodic variation in air pressure. For example it is unlikely (actually physically impossible) that a low and loud rumbling sound will come from a tiny bird. Therefore it is hardly surprising that most animal auditory systems are structured so that at the most basic level they can differentiate between sounds happening at different times and having different rates of vibration (frequencies). Without going into details, in most mammallian auditory systems sounds with different frequencies excite different groups of neurons (inner hair cells) attached to a membrane (the basilar membrane) that performs a type of frequency analysis; a given place along the basilar membrane responds more strongly to a sinusoidal vibration of a particular frequency. At the same time a given place along the basilar membrane will also react to other frequencies to a lesser degree and further mechanisms related to timing are used to better resolve these frequencies. Furthermore various structural features of the auditory cortex (the part of the brain that deals with understanding sound) are "tonotopically" organized. When excited by a sine wave of increasing frequency the excitation on the cortex moves continuously along a curved path. So there is something fundamental about frequency in the response of the brain to sound, right up to the highest level, the auditory cortex.

3.3.7 Frequency Domain Representations

A frequent and elegant approach to representing audio signals is to express them as weighted mixtures of elementary building blocks (basis functions) which can be thought of as simple prototypical signals. Then the representation essentially consists of the weights of the mixture. Audio signal representations computed using the same set of basis functions can be then analyzed and compared simply by considering the weights of the mixture. A common strategy in digital signal processing is to take advantage of a strategy called *superposition*. In superposition the signal being processed is decomposed into simple components, each component is processed separately and the results are reunited. Linear systems are typically required for superposition to work. As we have already discussed, in order to extract information from music audio signals the audio representation must somehow encode time and frequency. By analyzing the audio signal in small chunks the representation becomes dependent on time and by utilizing basis functions that correspond to different frequencies the weights of the mixture encode frequency.

There are four commonly used variants of frequency domain representations: the Fourier series, the Discrete Fourier Transform (DFT), the z-transform, and the classical Fourier transform. The intuition behind all of them is similar. The main idea is to express the signals of interest as weighted mixture of elementary signals; as you probably expect these will be some form of sinusoidal signals i.e. phasors. Armed with our knowledge of sinusoidal signals and how they can be viewed as phasors and notated using complex numbers we are ready to understand some of the most fundamental concepts in digital signal processing.

Coordinate systems

The weights of a mixture of elementary signals can be viewed as coordinates in a signal space spanned by these elementary signals in an analogous way to how the spatial coordinates of a point in space can be viewed as the weights that are needed to combine the unit-length basis vectors corresponding to each axis. For example a vector ν in three-dimensional space can be written as:

$$\nu = \nu_x \mathbf{x} + \nu_y \mathbf{y} + \nu_z \mathbf{z} \tag{3.19}$$

which can be interpreted as the fact that any vector in the three-dimensional space can be represented as a weighted combination of three parts: a vector in the x-direction of length ν_x, a vector in the y-direction of length ν_y and a vector in the z direction of length ν_z. The three unit length vectors in the coordinate directions $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are called a basis for the three dimensional space and the numbers ν_x, ν_y, ν_z are called the projections of vector ν onto the respective basis elements. We can denote the projection of a vector ν onto another vector u by $\nu < \nu, u >$ in which case:

$$\begin{aligned} \nu_x &= \langle \nu, \mathbf{x} \rangle \\ \nu_y &= \langle \nu, \mathbf{y} \rangle \\ \nu_z &= \langle \nu, \mathbf{z} \rangle \end{aligned}$$

The projection $\langle \nu, \upsilon \rangle$ is also called the inner product of ν and υ and can be defined as the sum of products of like coordinates (when the vector has coordinate values that are complex numbers the coordinates of the second vector need to be conjugated):

$$\langle \nu, \upsilon \rangle = \nu_x \upsilon_x^* + \nu_y \upsilon_y^* + \nu_z \upsilon_z^* \tag{3.20}$$

or more generally:

$$\langle \nu, \upsilon \rangle = \sum_i \nu_i \upsilon_i^* \tag{3.21}$$

The projection/inner product follows a distributive law and is symmetric which means that:

$$\langle \nu + \upsilon, \omega \rangle = \langle \nu, \omega \rangle + \langle \upsilon, \omega \rangle \tag{3.22}$$

$$\langle \nu, \upsilon \rangle = \langle \upsilon, \nu \rangle \tag{3.23}$$

Also notice that the projection of a vector onto itself (or equivalently the inner product of a vector with itself) is the square of the length of the vector (notice how the complex conjugate in the definition of the inner product is required to make this true when the coordinates are complex numbers):

$$\langle \nu, \nu \rangle = |\nu_x|^2 + |\nu_y|^2 + |\nu_z|^2 \tag{3.24}$$

The three coordinate basis vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are orthogonal to each other which means that their projections onto each other are zero.

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0$$
$$\langle \mathbf{x}, \mathbf{z} \rangle = 0$$
$$\langle \mathbf{y}, \mathbf{z} \rangle = 0$$

When the basis vectors are mutually orthogonal like this the basis is called an *orthogonal basis*. So basically in order to create an orhtogonal coordinate system we need two essential components: **a projection operator**, and an **orthogonal basis**. We are now ready to use these concepts in order to better understand frequency domain representations which we will define by specifying appropriate projection operators and orthogonal basis vectors which in this case will be some form of phasors.

Frequency Domain Representations as coordinate systems

Suppose we want to represent period signals that are functions of a continuous time variable t defined in the range $0 \leq t \leq T$ (think of the signal as repeating

outside this range). Even though it might seem strange we can think of every possible value of t in the range 0 to T as a coordinate. In this case instead of having a finite number of dimensions we have an infinite number of them. We can then generalize the definition of the projection operator (or inner product) to use a continuous integral:

$$\langle f, g \rangle = \frac{1}{T} \int_0^T f(t) g^*(t) dt \tag{3.25}$$

The basis elements should also be periodic signals of a continuous time variable t and as you probably have guessed they are the phasors that have period T:

$$e^{jtk2\pi/T}, \quad k = \ldots, -1, 0, 1, 2 \ldots \tag{3.26}$$

Notice that negative frequencies are included in order to represent real functions of t. Geometrically the phasor for $k = 1$ will complete one circle in time T, the phasor for $k = 2$ will complete two circles, etc. The negative frequencies correspond to phasors moving at the same speed but clockwise instead of counterclockwise. It is also straightforward to show that this basis is orthogonal. It is important to notice that using this representation the coordinate system has been radically transformed from one that is continuously indexed (time) to one that is discretely indexed (the phasor basis indexed by integers). Although there are some mathematic restrictions it is remarkable that this works.

Based on these two definitions of projection operator and basis functions we can now express any periodic function in terms of the basis:

$$f(t) = \sum_{-\infty}^{\infty} C_k e^{jkt2\pi/T} \tag{3.27}$$

The complex coefficient C_k can be thought of as the coordinate in the "direction" of the phasor $e^{jkt2\pi/T}$ or the amount of the frequency $k2\pi/T$. Equation 3.27 is called the **Fourier Series** of $f(t)$ and the values C_K are called the **spectrum** of the periodic signal $f(t)$. To find the Fourier coefficients C_k we can simply project $f(t)$ on the kth basis element (having a bad memory I always rely on the definition of the inner product with the complex conjugate to determine the sign of the exponent in frequency transforms):

$$C_k = \langle f(t), e^{jkt2\pi/T} \rangle = \frac{1}{T} \int_0^T f(t) e^{-jkt2\pi/T} dt \tag{3.28}$$

Notice that when $f(t)$ is a real (as is the case for audio signals) there is a very simple relationship between the coefficients for negative and positive frequencies:

$$C_k = C^*_{-k} \tag{3.29}$$

3.3.8 Discrete Fourier Transform

The Fourier Series representation transforms periodic signals from a continuous time domain to an infinite discrete frequency domain. Therefore it is not directly usable when processing digital signals. The transform that is needed is called the Discrete Fourier Transform (DFT). It transforms a finite discrete time domain signal (or an array of numbers in computer parlance) to a finite discrete frequency domain signal (also an array of numbers). The DFT is arguably the most widely used transform in digital signal processing and as an extension in MIR. In part this because of a very fast algorithm for computing it that is called the Fast Fourier Transform (FFT). Even though the technique was first discovered by Gauss in 1805 it became widely used in its current form through the work of Cooley and Tukey []. In the published literature frequently the terms DFT and FFT are used interchangeably but it is good to keep in mind the the DFT refers to the transform whereas the FFT refers to a fast algorithm for computing it. Similarly to how the Fourier Series was presented we define the DFT by specifying a projection operator and a set of basis functions. As probably expected the projection operator is the standard inner product between vectors of complex numbers:

$$\langle x, y \rangle = \sum_i x_t y^*_t \tag{3.30}$$

Due to aliasing, frequencies of discrete-time signals are equivalent modulo the sampling frequency.

Arguably the most basic elementary basis function is the sampled sinusoid which forms the basis of the Discrete Fourier Transform and the Short Time Fourier Transform (STFT). The STFT forms the foundation of most proposed algorithms for the analysis of audio signals and music. So our description of audio representations begins with the sound of a sinusoid.

Audio Examples Source Code (Marsyas/Matlab) Figure

3.3.9 Sampling and Quantization

3.3.10 Discrete Fourier Transform

3.3.11 Linearity, propagation, resonances

3.3.12 The Short-Time Fourier Transform

Music-humans-language-mental models-abstraction-better time modeling scales

MPEG-7 standard

"feature" is a distinctive characteristic of the data which signifies something to somebody.

3.3.13 Wavelets

3.4 Perception-informed Representations

3.4.1 Auditory Filterbanks

3.4.2 Perceptual Audio Compression

3.5 Source-based Representations

LPC

3.6 Further Reading

The theory behind the audio representations used in music information retrieval forms the field of Digital Signal Processing (DSP). There are many excellent DSP books. The following list consists of a few of my own personal preferences rather

than any attempt at a comprehensive catalog. The DSP Primer by Ken Steiglitz [**?**] is my favorite beginner book as it stresses intuition and has a strong emphasis on music. The classic book by Oppenheim and Schafer [**?**] is the essential reference for anyone needing a deeper understanding of signal processing.

Digital Signal Processing: A Practical Guide for Engineers and Scientists [Paperback] Steven Smith (Author)

Paperback: 650 pages Publisher: Newnes; Book and CD ROM edition (November 6, 2002) Language: English ISBN-10: 075067444X ISBN-13: 978-0750674447 Product Dimensions: 9.9 x 7.2 x 1.2 inches Shipping Weight: 2.6 pounds

Understanding Digital Signal Processing (2nd Edition) [Hardcover] Richard G. Lyons (Author) Hardcover: 688 pages Publisher: Prentice Hall; 2 edition (March 25, 2004) Language: English ISBN-10: 0131089897 ISBN-13: 978-0131089891

Hardcover: 1120 pages Publisher: Prentice Hall; 3 edition (August 28, 2009) Language: English ISBN-10: 0131988425 ISBN-13: 978-0131988422 Product Dimensions: 9.3 x 7.5 x 1.7 inches Shipping Weight: 4.2 pounds (View shipping rates and policies)

Discrete-Time Signal Processing (3rd Edition) (Prentice Hall Signal Processing) [Hardcover] Alan V. Oppenheim (Author), Ronald W. Schafer (Author)

Problems

1. Show that a sinusoidal signal with any phase angle satisfies equation 3.5

2. Show that the basis fuctions for the Fourier Series are orthogonal using the appropriate definition of an inner product.

3. Plot the DFT magnitude response in dB (using a DFT size of 2048) of a sine wave that is exactly equal to the center frequency of DFT bin 600. On the same plot overlap the DFT magnitude response in dB of a sine wave with frequency that would correspond to bin 600.5 (in a sense falling between the crack of two neigh- boring frequency bins). Finally on the same plot overlap the DFT magnitue rsponse in dB of the second sine wave (600.5) windowed by a hanning window of size 2048. Write 2-3 brief sentences describing and explaning the three plots and what is the effect of windowing to the mangitude response for these inputs.

Chapter 4

Music Representations

Music is an ephemeral activity and for most of human history it only existed while it was created. The only way it was represented was in the mental memories of musicians. Although such mental representations can actually be quite abstract and effective in terms of retrieval, their workings are a mystery and not easity understood through introspection. For example, a musician might easily recognize a tune from a badly rendered melodic fragment sung by an audience member or be able to play from memory a highly complex piece of music. However in most cases it is impossible or very hard to describe how these tasks are accomplished. The use of external symbolic representations to communicate and store information was a major technological advancement in human history. Several writing systems were developed codifying different types of information the most well known being logistical/accounting information, concepts/words (hieroglyphics) and phonemes/sounds (alphabet). Gradually symbolic representations for codifying different aspects of music were developed with western common music notation being probably the most well known and used example today. Musicians spend many hours learning and studying how to codify music in scores and how to perform music written as a score. Even though a full music education is not required to work on music information retrieval a basic understanding of how scores codify music is essential in comprehending several tasks that have been investigated in MIR. This chapter is an overview of the basic music theory, concepts and abstractions involved when representing music symbolically. For

readers with a formal music background this material is probably familiar and should be skipped as needed.

Music throughout most of human history was trasmitted orally and only existed ephemerally at the time and place it was performed. Language and poetry were similarly orally transmitted until the development of writing which allowed linguistic information to be preserved over time and space using physical objects such as clay tablets. Cuniform script is one of the earliest forms of writing and it is perhaps not surprising that one of the earliest examples of music notation is in cuniform dated to 2000 BC [**?**].

Over time several cultures developed a variety of music notation systems. To some extent all these notation systems can be considered as mnemonic aids for performers differing in the type of information they encode. For example many notation systems are designed for specific instruments such as guitar tablature notation which encodes information about what string to play and when. The structured symbolic representation used to notate music is called a musical **score**. The symbols on the score correspond to various types of acoustic events and encode information about the gestures required to produce them.

Music notation gradually evolved from a simple mnemonic aid, and transformed the way music was produced, consumed and distributed. Through scores music could be transmitted across space and time. Without notation we would not have been able to listen to the music written by J.S. Bach today and J. S. Bach would not have been able to study Italian composers during his time. In fact, the profession of a composer only became possible with music notation. Finally, musical scores were and still are instrumental in the field of musicology and music theory which, among other things, study the process of music composition and the rules (and exceptions) that govern it.

Music scores can be viewed as a set of instructions to produce a particular musical output. Viewed that way they are similar to computer programs. This analogy is not as far fetched as it might seem at first glance. Common music notation has conventions for what are essentially looping constructs and goto statements. Early mechanical calculators were influenced by the automated Jackard looms which in turn were influenced by player piano mechanisms. So with some speculation one can consider music notation a proto programming language making an interesting prelude to the much later development of music information retrieval techniques.

The "dimensions" of time and frequency are fundamental in understanding music from cultures from around the world and throughout history. Pitch refers to the subjective perception of frequency by humans. It is also important to keep in

mind that a periodic sound will have a clear pitch only if it has sufficient duration. Much of our knowledge of pitch perception comes from psychophysics, the psychology discipline that investigates the relationship between physical stimuli and their subjective perception. For example an important concept in psychophysics is the idea of "just noticable difference" (JND) which is the difference in physical stimuli that subjects can not detect reliably. In a typical experimental setup for pitch perception the subject is played two sine waves of different frequencies and is asked to say whether they are the same or not. Based on similar experiments it can be shown that the total number of perceptible pitch steps in the range of human hearing is approximately 1400 (several factors including individual differences, intensity, duration and timbre can affect this number).

Categorical perception occurs when the continuous stimuli that reach our sense organs is sorted out by our brains into distinct categories whose members come to resemble more one another than they resemble members of the other categories. The canonical example of categorical perception is how we perceive colors. In physical terms colors differ only in their wavelength which gets shorter across the spectrum of visible colors. However we perceive discrete categories such as red, orange and green. Similar categorical perception can be observed in speech and more importantly for our purposes in the perception of pitch.

Many music cultures utilize some form of division of the pitch continuum in discrete categories. Some of these divisions are more formalized while others are more subjective and vary among culture, instrument and performers. An interval between two sounds is their spacing in pitch. Most humans have categorical perception of pitch intervals. The identity of a melody (sequence of notes) like "Twinkle, Twinkle Little Star" is based on a sequence of pitch intervals and is, to a large extent, not affected by the frequency of the starting note. The most fundamental interval is the "octave" or a doubling in frequency which is recognized by most music traditions. Tuning systems are formalized theories that describe how categorical pitch intervals are formed. All tuning systems are affected by the physics of sound production and perception but they are also affected by music traditions and tuning pecularities of the instruments used in those traditions. Pitch intervals are perceived as ratios of frequencies so that an octave above 440 Hz (the frequency used today for tuning in most western classical music) would be 880 Hz and an octave above 880 Hz would be 1760 Hz. Most tuning systems can be viewed as different ways of subdividing the octave (the interval corresponding to a frequency ratio of two). Tuning systems and their relations with timbre and acoustics are a fascinating topic however for the purposes of this book we will only

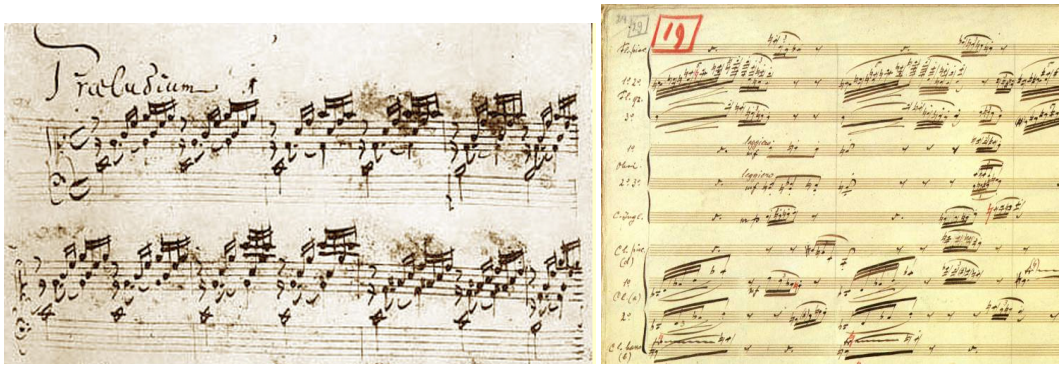


Figure 4.1: Score Examples

consider two well known variants: Pythagorean tuning and Equal Temperement. More details about tuning systems can be found in Sethares [**?**].

In this chapter, the basic conventions and concepts used in Western common music notation are introduced as it is the most widely used notation system today. The chapter also serves as a compact introduction to basic music theory for readers who do not have a formal music background. When reading MIR literature it is common to encounter terms from music theory and have figures with music score examples so some basic knowledge of music theory is useful. Music theory and notation are complex topics that can take many years to master and in this chapter we barely scratch the surface of the subject. I hope that this exposition motivates you to get some formal music training. This overview of music theory and notation is followed by a presentation of digital formats for symbolic music representation.

4.1 Common music notation

Figure 4.1 shows two examples of hand written scores. The score on the left is the beginning of the well-known prelude in C by J. S. Bach and the one on the right is an excerpt of the Firebird Suite by Igor Stravinsky. Even though the underlying music is very different and the manuscripts were written more than 200 years apart, the notation symbols and conventions are very similar.

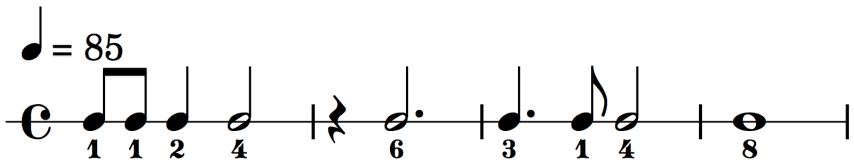


Figure 4.2: Rhythm notation

4.1.1 Notating rhythm

Rhythm is encoded using combinations of symbols (circles, dots and stems) that indicate relative duration in terms of a theoretical regular underlying pulse. In many music styles there are regular groupings of rhythmic units called measures and any particular rhythmic event including a regular pulse can be expressed as a subdivision of the measure. For example a lot of music has 4 subdivisions in each measure and each one is called a quarter note. A whole note lasts as much as 4 quarter notes. A finer subdivision is in eigth notes which as you might guess last half the duration of a quarter note. Figure 4.2 shows a couple of examples of rhythmic symbols with numbers under them indicating the number of eigth notes they correspond to. Notice the use of stems, circle filling, and dots as duration modifiers. The encoding of rhythmic durations is relative and the same score can be performed faster or slower depending on the speed of the underlying regular pulse. This can be specified by tempo indicators such as the 85 in Figure 4.2 which indicates that there should be 85 quarter notes (beats) in each minute. When the tempo is specified then the relative durations indicated by the symbols can be converted to absolute durations.

define beat, tatum, tactus

4.1.2 Notating pitch

4.1.3

4.2 MIDI

4.3 Notation formats

MusicXML, Guido, Lilypond

4.4 Music Theory

intervals, scales, chords, cadences

4.5 Graphical Score Representations

4.6 MIDI

4.7 MusicXML

[37] [90] [**?**] [49] [64]
[18]
[19]

Chapter 5

Feature Extraction

Audio feature extraction forms the foundation for any type of music data mining. It is the process of distilling the huge amounts of raw audio data into much more compact representations that capture higher level information about the underlying musical content. As such it is much more specific to music than the data mining processes that typically follow it.

A common way of grouping audio features (or descriptors as they are sometimes called) is based on the type of information they are attempting to capture [101]. On an abstract level one can identify different high level aspects of a music recording. The hierarchical organization in time is refered to as *rhythm* and the hierarchical organization in frequency or pitch is refered to as *harmony*. Timbre is the quality that distinguishes sounds of the same pitch and loudness generated by different methods of sound production. We will use the term *timbral texture* to refer to the more general quality of the mixture of sounds present in music that is independent of *rhythm* and *harmony*. For example the same piece of notated music played at the same tempo by a string quartet and a saxophone quartet would have different *timbral texture* characteristics in each configuration.

In this chapter various audio feature extraction methods that attempt to capture these three basic aspects of musical information are reviewed. We begin by examining monophonic fundamental frequency (or pitch) estimation as it is one of the most basic audio features one can calculate, has interesting applications, and can help motivate the discussion of other features. Some additional audio features that cover other aspects of musical information are also briefly described. The

chapter ends with a short description of audio genre classification as a canonical case study of how audio featue extraction can be used as well as some references to freely available software that can be used for audio feature extraction. Audio feature extraction is a big topic and it would be impossible to cover it fully in this chapter. Although our coverage is not complete we believe we describe most of the common audio feature extraction approaches and the bibliography is representative of the "state of the art" in this area in 2013.

5.1 Monophonic pitch estimation

As we saw in Chapter 12 music notation systems typically encode information about discrete pitch (notes on a piano) and timing. When reading literature in music information retrieval and associated research areas the term *pitch* is used in different ways which can result in some confusion.

5.1.1 Terminology

In this book I have tried to be more precise by using the following terms:

Perceptual Pitch: is a perceived quality of sound that can be ordered from "low" to "high".

Musical Pitch: refers to a discrete finite set of perceived pitches that are played on musical instruments

Measured Pitch: is a calculated quantity of a sound using an algorithm that tries to match the perceived pitch.

Fundamental frequency:

The pitch extraction methods described in this chapter are designed for *monophonic* audio signals. *Monophoning* pieces of music are ones in which a single sound source (instrument or voice) is playing and only one pitch is heard at any particular time instance. *Polyphonic* refers to pieces of music in which multiple notes of the same instrument are heard simultanesouly as in a piece of piano music or pieces with multiple instruments playing such as a symphony or a pop ballad. Polyphonic pitch extraction and additional aspects of monophonic pitch extraction are coverd in Chapter **??**.

5.1.2 Psychoacoustics

A lot of what we know about perceptual pitch comes from the field of Psychoacoustics, which is the scientific study of sound perception. The origins of psychoacoustics can be traced all the way back to ancient Greece and Pythagoras of Samos. Pythagoras noticed that melodies consisted of intervals (sequences of two pitches) and was able to establish a connection between the intervals used by musicians and physical measurable quantities. More specifically he used an instrument he designed called the monochord which consisted of a single string with a movable bridge. He experimentally established that the intervals musicans used corresponded to dividing the string into integer ratios using the movable bridge.

Modern psychoacousticians also try to probe the connection between physical measurements and perception of course using more sophisticated tools. For example, frequently we are interested in the limits of perception such as what is the highest (or lowest) frequency one can hear. This can be established by playing sine waves of increasing frequency until the listener can not hear them anymore. Young listers can hear frequencies up to approximately 20000 Hz with the upper limit getting lower with age. Today it is relatively straightforward to do simple psychoacoustic experiments using a computer and headphones (some ideas can be found in the problems at the end of this chapter). Similar limits can be established with loudness. Another important concept in pyschoacoustics is the Just Noticeable Difference (JND) which refers to the minimum amount a physical quantity needs to change in order for the change to be perceived. For example if I play a sine wave of a particular frequency followed by another one of a slightly higher frequency how much higher does the second one need to be in order for the change to be perceived.

Through simple psychoacoustic experiments one can establish two important findings:

- We are able to perceive the frequencies of sine waves as ordered from low to high
- More complex periodic sounds such as those produced by musical instruments are also perceived as ordered from low to high and can be matched with corresponding sine waves in terms of their perceived pitch

These findings motivate a very common experimental paradigm for studying pitch perception in which the listener hears a complex sound and then adjusts the

frequency of sine wave generator until the sine wave and the compelx sound are perceived as having the same pitch. The frequency of the adjusted sine wave is defined as the perceptual or perceived pitch of the sound. For a large number of musical sounds the perceived pitch corresponds to the fundamental frequency of the sound i.e the lowest frequency with significant energy (or peak) present in the sound when performing Fourier analysis.

5.1.3 Musical Pitch

5.1.4 Time-Domain Pitch Estimation

5.1.5 Frequency-Domain Pitch Estimation

5.1.6 Perceptual Pitch Estimation

5.2 Timbral Texture Features

Features representating timbral information have been the most widely used audio features and ones that have so far provided the best results when used in isolation. Another factor in their popularity is their long history in the area of speech recognition. There are many variations in timbral feature extraction but most proposed systems follow a common general process. First some form of time-frequency analysis such as the STFT is performed followed by summarization steps that result in a feature vector of significantly smaller dimensionality. A similar approach can be used with other time-frequency represenations.

5.2.1 Spectral Features

Spectral features are directly computed on the magnitude spectrum and attempt to summarize information about the general characteristics of the distribution of energy across frequencies. They have been motivated by research in timbre perception of isolated notes of instruments [41]. The spectral centroid is defined

as the center of gravity of the spectrum and is correlated with the perceived "brightness" of the sound. It is defined as:

$$C_n = \frac{\sum_{k=0}^{N-1} |X[k]|_n k}{\sum_{k=0}^{N-1} k} \tag{5.1}$$

where n is the frame number to be analyzed, k is the frequency bin number and $|X(k)|_n$ is the corresponding magnitude spectrum.

The spectral rolloff is defined as the frequency R_n below which 85% of the energy distribution of the magnitude spectrum is concentrated:

$$\sum_{n=0}^{R_n-1} = 0.85 * \sum_{n=0}^{N-1} |X[k]|_n \tag{5.2}$$

5.2.2 Mel-Frequency Cepstral Coefficients

The Mel-Frequency Cepstral Coefficients MFCC [22] are the most common representation used in automatic speech recognition systems and have been frequently used for music modeling. Their computation consists of 3 stages: 1) Mel-scale filterbank 2) Log energy computation 3) Discrete Cosine Transform.

A computationally inexpensive method of computing a filterbank is to perform the filtering by grouping STFT bins using triangular windows with specific center frequencies and bandwidths. The result is a single energy value per STFT frame corresponding to the output of each subband. The most common implementation of MFCC is calculated using 13 linearly spaced filters separated by 133.33 Hz between their center frequencies, followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency) resulting in 40 filterbank values for each STFT frame.

The next step consists of computing the logarithm of the magnitude of each of the filterbank outputs. This can be viewed as a simple step of dynamic compression, making feature extraction less sensitive to variations in dynamics.

The final step consists of reducing the dimensionality of the 40 filterbank outputs by performing a Discrete Cosine Transform (DCT) which is similar to the Discrete Fourier Transform but uses only real numbers. It expresses a finite set of values in terms of a sum of cosine functions oscillating at difference frequencies. The DCT is used frequently in compression applications because for typical signals of intererst in tends to concentrate most of the signal information in few of the lower frequency components and therefore higher frequency components can

be discarded for compression purposes. In the "classic" MFCC implementation the lower 13 coefficients are retained after transforming the 40 logarithmically compressed Mel filterbank outputs.

[72]

5.2.3 Other Timbral Features

There have been many other features proposed to describe short-term timbral information. In this subsection we briefly mention them and provide citations to where the details of their calculation can be found. Time domain zero-crossings can be used to measure how noisy the signal is, and also somewhat correlate to high frequency content [101, 10]. Spectral bandwidth [10, 30, 73] and octave based spectral contrast [53, 73] are other features that attempt to summarize the magnitude spectrum. The spectral flatness measure and spectral crest factor [1] are low level descriptors proposed in the context of the MPEG-7 standard [15].

[71]

Daubechies Wavelet Coefficient Histogram is a technique for audio feature extraction based on the Discrete Wavelet Transform (DWT) [67]. It is applied in 3 seconds of audio using the db_8 Daubechies wavelet filter [21] with seven levels of decomposition. After the decomposition, the histograms of the coefficients in each band are calculated. Finally each histogram is characterized by its mean, variance and skewness as well as the subband energy defined as the mean of the absolute coefficient values. The result is 7 (subbands) * 4 (quantities per subband) = 28 features.

One of the major innovations that enabled the explosive growth of music represented digitally is perceptual audio compression [83]. Audio compression is frequently used to reduce the high data requirements of audio and music. Audio data does not compress well using generic data compression algorithms so specialized algorithms have been developed. The majority of audio coding algorithms are lossy meaning that the original data can not be reconstructed exactly from the compressed signal. They are frequently based on some form of time-frequency representation and they achieve compression by allocating different number of bits to different parts of the signal. One of the key innovations in audio coding is the use of psychoacoustics (i.e the scientific study of sound perception by humans) to guide this process so that any artifacts introduced by the compression are not perceptible.

There has been some interest in computing audio features directly in the compressed domain as part of the decompression process. Essentially this takes advantage of the fact that a type of time-frequency analysis has already been performed for encoding purposes and it is not necessary to repeat it after decompression. The type of features used are very similar to the ones we have described except for the fact that they are computed directly on the compressed data. Early works mainly focused on the MPEG audio compression standard and the extraction of timbral texture features [100, 88]. More recently the use of newer sparse overcomplete representations as the basis for audio feature extraction has been explored [28] covering timbral texture, rhythmic content and pitch content.

5.2.4 Temporal Summarization

The features that have been described so far characterize the content of a very short segment of music audio (typically around 20-40 milliseconds) and can be viewed as different ways of summarizing frequency information. Frequently a second step of temporal summarization is performed to characterize the evolution of the feature vectors over a longer time frame of around 0.5 to 1 seconds. We can define a "texture" window $T_M[n]$ of size M corresponding to analysis frame n as the sequence of the previous $M-1$ computed feature vectors including the feature vector corresponding to n.

$$T_M[n] = F[n-M+1]...F[n] \tag{5.3}$$

Temporal integration is performed by summarizing the information contained in the texture window to a single feature vector. In other words the sequence of M past feature vectors is "collapsed" to a single feature vector. At the one extreme, the texture window can be advanced by one analysis frame at a time in which case the resulting sequence of temporally summarized feature vectors has the same sampling rate as the original sequence of feature vectors [101]. At the other extreme temporal integration can be performed across the entire length of the audio clip of interest resulting in a single feature vector representing the clip (sometimes such features are termed song-level [75] or aggregate features [10]). Figure 5.1 shows schematically a typical feature extraction process starting with a time-frequency representation based on the STFT, followed by the calculation of MFCC (frequency) summarization and ending with summarization of the resulting feature vector sequence over the texture window.

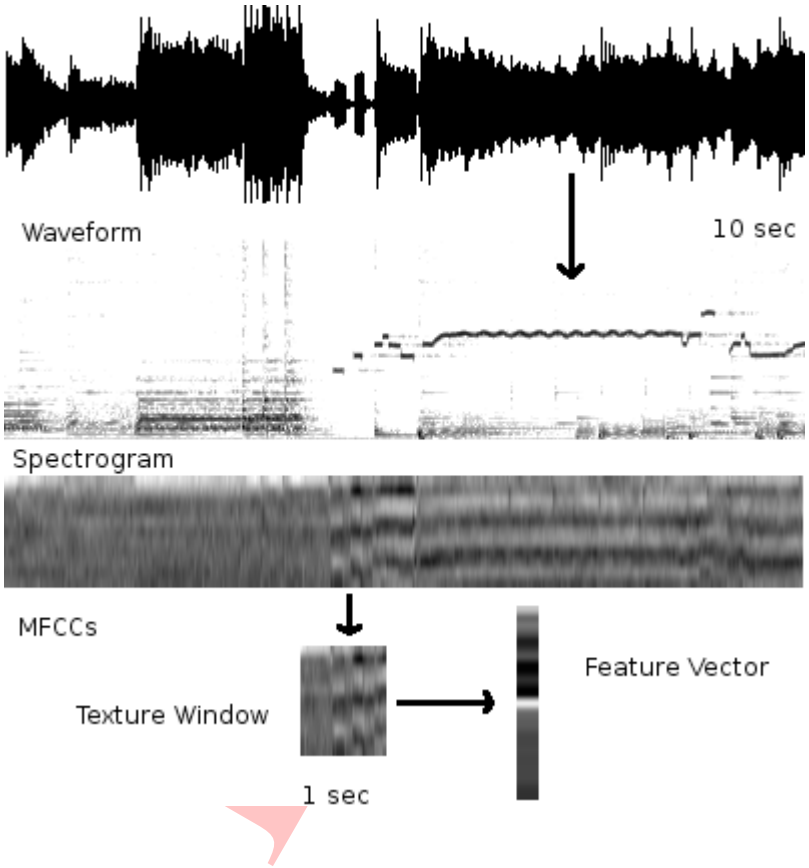


Figure 5.1: Feature extraction showing how frequency and time summarization with a texture window can be used to extract a feature vector characterizing timbral texture

The is no consistent terminology describing temporal summarization. Terms that have been used include: dynamic features [87, 76] , aggregate features [10], temporal statistics [30], temporal feature integration [78], fluctuation patterns [89], and modulation features [63]. More recently the term pooling has also been used [**?**].

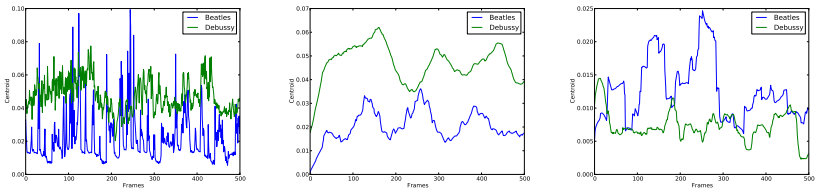
Statistical moments such as the mean, standard deviation and the covariance matrix can be used to summarize the sequence of feature vectors over the texture window into a single vector. For example a common approach is to compute the means and variances (or standard deviations) of each feature over the texture window [101]. Figure 5.2 shows the original trajectory of spectral centroids over a texture window as well as the trajectory of the running mean and standard deviation of the spectral centroid for two pieces of music.

Another alternative is to use the upper triangular part of the covariance matrix as the feature vector characterizing the texture window [75]. Such statistics capture the variation of feature vectors within a texture window but do not directly represent temporal information.

Another possibility is to utilize techniques from multivariate time-series modeling to characterize the feature vector sequence that better preserve temporal information. For example multivariate autoregressive models have been used to model temporal feature evolution [78]. The temporal evolution of the feature vectors can also be characterized by performing frequency analysis on their trajectories over time and analyzing their periodicity characteristics. Such modulation features show how the feature vectors change over time and are typically calculated at rates that correspond to rhythmic events. Any method of time-frequency analysis can be used to calculate modulation features but a common choice is the Short-Time Fourier Transform (STFT).

As a representative example of calculating modulation frequencies we briefly describe the approach proposed by McKinnery and Breebart [76]. A standard MFCC computation is performed resulting in a sequence of 64 feature vectors each with 13 dimensions characterizing a texture window of 743 milliseconds of audio. A power spectrum of size 64 using a DFT is calculated for each of the 13 coefficients/dimesions resulting in 13 power spectra which are then summarized in 4 frequency bands ($0Hz$, $1-2Hz$, $3-15Hz$, $20-43Hz$) that roughly correspond to musical beat rates, speech syllabic rates and modulations contributing to perceptual roughness. So the final representations for the $13x64$ matrix of feature values of the texture window is $4x13 = 52$ dimensions.

A final consideration is the size of the texture windows and the amount of overlap between them. Although the most common approach is to use fixed



(a) Centroid

(b) Mean Centroid over Texture Window

(c) Standard Deviation of Centroid over Texture Window

Figure 5.2: The time evolution of audio features is important in characterizing musical content. The time evolution of the spectral centroid for two different 30-second excerpts of music is shown in (a). The result of applying a moving mean and standard deviation calculation over a texture window of approximately 1 second is shown in (b) and (c).

window and hop sizes there are alternatives. Aligning texture windows to note events can provide more consistent timbral information [106]. In many music analysis applications such as cover song detection it is desired to obtain an audio feature representation that is to some extent tempo-invariant. One way to achieve this is using so-called beat-synchronous features which as their name implies are calculated using estimated beat locations as boundaries [29]. Although more commonly used with features that describe pitch content they can also be used for timbral texture modeling.

5.2.5 Song-level modeling

Frequently in music data mining the primary unit of consideration is an entire track or excerpt of a piece of music. The simplest and most direct type of representation is a single feature vector that represents the entire piece of music under consideration. This is typically accomplished by temporal summarization techniques such as the ones described in the previous section applied to the entire sequence of feature vectors. In some case two stages of summarization are performed: one

at the texture window level and one across the song [101]. The efffectiveness of different parameter choices and temporal summarization methods has also been explored [10].

In other music data mining problems the entire sequence of feature vectors is retained. These problems typically deal with the internal structure within a music piece rather than the relations among a collection of pieces. For example in audio segmentation [31, 98] algorithms the locations in time where the musical content changes significantly (such the transition from a singing part to an electric guitar solo) are detected. Audio structure analysis goes a step further and detects repeating sections of a song and their form such as the well known AABA form [20]. Sequence representations are also used in cover song detection [29].

A representation that is frequently utilized in structure analysis is the self similarity matrix. This matrix is calculated by calculating pair-wise similarities between feature vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ derived from audio analysis frames i and j.

$$s(i,j) = sim(\mathbf{v_i}, \mathbf{v_j}) \tag{5.4}$$

where sim is a function that returns a scalar value corresponding to some notion of similarity (or symmetrically distance) between the two feature vectors. Music is generally self-similar with regular structure and repetitions which can be revealed through the self similarity matrix.

Figure 5.3 shows an example of a self-similarity matrix calculated for a piece of HipHop/Jazz fusion using simply energy contours (shown to the left and bottom of the Matrix). The regular structure at the beat and measure level as well as some sectioning can be observed.

The final variation in song-level representations of audio features is to model each music piece as a distribution of feature vectors. In this case a parametric distribution form (such as a Gaussian Mixture Model [3]) is assumed and its parameters are estimated from the sequence of feature vectos. Music data mining tasks such as classification are performed by considering distance functions between distributions typically modeled as mixture models such as the KL-divergence or Earth Mover Distance [75, 52].

5.3 Rhythm Features

Automatically extracting information related to rhythm is also an important component of audio MIR systems and has been an active area of research for over 20

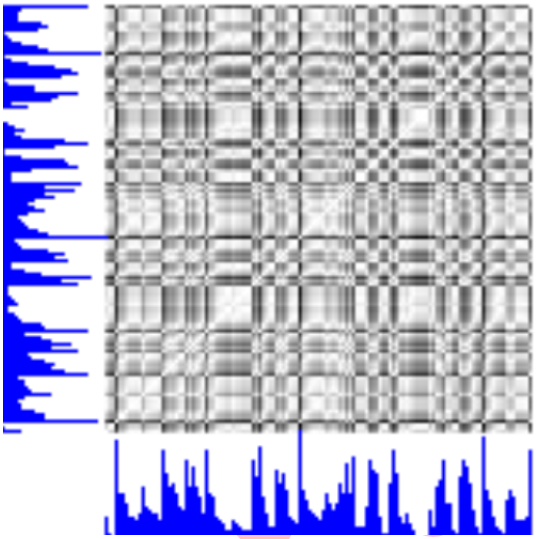


Figure 5.3: Self Similarity Matrix using RMS contours

years. A number of different subproblems within this area have been identified and explored. The most basic approach is finding the average tempo of the entire recording which can be defined as the frequency with which a human would tap their foot while listening to the same piece of music. The more difficult task of beat tracking consists of estimating time-varying tempo (frequency) as well as the locations in time of each beat (phase). Rhythmic information is hierarchical in nature and tempo is only one level of the hierarchy. Other levels frequently used and estimated by audio MIR systems are tatum (defined as the shortest commonly occurring time interval), beat or tactus (the preferred human tapping tempo), and bar or measure. For some MIR applications such as automatic classification it is possible to use a a representation that provides a salience value for every possible tempo e.g., the beat histograms described in [101]. Rhythm analysis approaches can be characterized in different ways. The first and most important distinction is by the type of input: most of the earlier beat tracking algorithms used a symbolic representation while audio signals have been used more recently. Symbolic algorithms can still be utilized with audio signals provided an intermediate transcription step is performedtypically audio onset detection. Another major distinction between the algorithms is the broad approach used which includes rulebased, autocorrelative, oscillating filters, histogramming, multiple agent, and

probabilistic. A good overview of these approaches can be found in Chapter 4 of Klapuri and Davy [43].

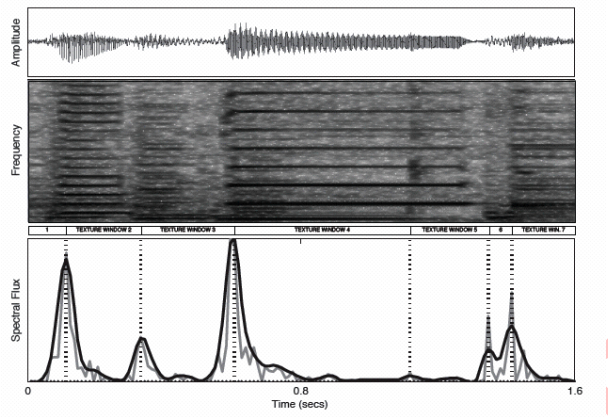


Figure 5.4: The top panel depicts the time domain representation of a fragment of a polyphonic jazz recording, below which is displayed its corresponding spectrogram. The bottom panel plots both the onset detection function SF(n) (gray line), as well as its filtered version (black line). The automatically identified onsets are represented as vertical dotted lines.

5.3.1 Onset Strength Signal

Frequently, the first step in rhythm feature extraction is the calculation of the onset strength signal. The goal is to calculate a signal that has high values at the onsets of musical events. By analyzing the onset strength signal to detect common recurring periodicities it is possible to perform tempo induction, beat tracking as well as other more sophisticated forms of rhythm analysis. Other names used in the literature include onset strength function and novelty curve.

The onset detection algorithm described is based on a recent tutorial article [24], where a number of onset detection algorithms were reviewed and compared on two datasets. Dixon concluded that the use of a spectral flux detection function for onset detection resulted in the best performance versus complexity ratio.

The spectral flux as an onset detection function is defined as :

$$SF(n) = \sum_{k=0}^{N/2} H(|X(n,k)| - |X(n-1,k)|) \tag{5.5}$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function, $X(n,k)$ represents the k-th frequency bin of the n-th frame of the power magnitude (in dB) of the short time Fourier Transform, and N is the corresponding Hamming window size. The bottom panel of Figure 5.4 plots the values over time of the onset detection function SF(n) for an jazz excerpt example.

The onsets are subsequently detected from the spectral flux values by a causal peak picking algorithm, that attempts to find local maxima as follows. A peak at time $t = \frac{nH}{fs}$ is selected as an onset if it satisfies the following conditions:

$$SF(n) \geq SF(k) \quad \forall k : n - w \leq k \leq n + w \tag{5.6}$$

$$SF(n) > \frac{\sum_{k=n-w}^{k=n+w} SF(k)}{mw + w + 1} \times thres + \delta \tag{5.7}$$

where $w = 6$ is the size of the window used to find a local maximum, $m = 4$ is a multiplier so that the mean is calculated over a larger range before the peak, $thres = 2.0$ is a threshold relative to the local mean that a peak must reach in order to be sufficiently prominent to be selected as an onset, and $\delta = 10^{-20}$ is a residual value to avoid false detections on silent regions of the signal. All these parameter values were derived from preliminary experiments using a collection of music signals with varying onset characteristics.

As a way to reduce the false detection rate, the onset detection function SF(n) is smoothed (see bottom panel of Figure 5.4),using a Butterworth filter defined as:

$$H(z) = \frac{0.1173 + 0.2347z^{-1} + 0.1174z^{-2}}{1 - 0.8252z^{-1} + 0.2946z^{-2}} \tag{5.8}$$

In order to avoid phase distortion (which would shift the detected onset time away from the SF(n) peak) the input data is filtered in both the forward and reverse directions. The result has precisely zero-phase distortion, the magnitude is the square of the filter's magnitude response, and the filter order is double the order of the filter specified in the equation above.

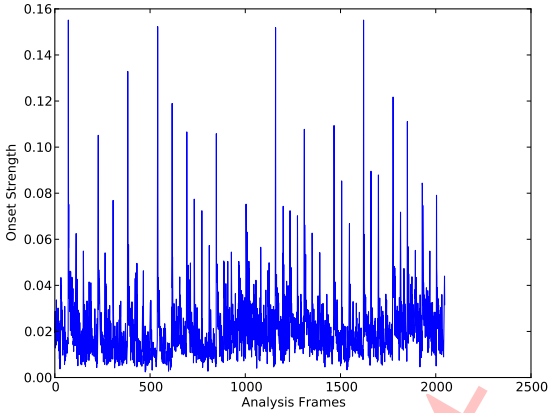


Figure 5.5: Onset Strength Signal

5.3.2 Tempo Induction and Beat Tracking

Many pieces of music are structured in time on top of an underlying semi-regular sequence of pulses frequently accentuated by percussion instruments especially in modern popular music. The basic tempo of a piece of music is the rate of musical beats/pulses in time, sometimes also called the "foot-tapping" rate for obvious reasons. Tempo induction refers to the process of estimating the tempo of an audio recording. Beat tracking is the additional related problem of locating the positions in time of the associated beats.

In this section we describe a typical method for tempo induction as a representative example and provide pointers to additional literature in the topic. The first step is the calculation of the onset strength signal as described above. Figure 5.5 shows an example of an onset strength signal for a piece of HipHop/Jazz fusion showing periodicities at the beat and measure level. The autocorrelation of the onset strength signal will exhibit peaks at the lags corresponding to the periodicities of the signal. The autocorrelation values can be warped to form a beat histogram which is indexed by tempos in beats-per-minute (BPM) and has values proportional to the sum of autocorrelation values that map to the same tempo bin. Typically either the highest or the second highest peak of the beat histogram corresponds to the tempo and can be selected with peak picking heuristics. Figure 5.6 shows two example beat histograms from 30 second clips of HipHop Jazz

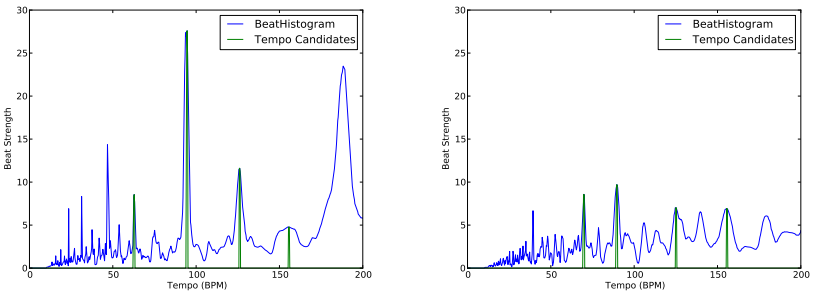


Figure 5.6: Beat Histograms of HipHop/Jazz and Bossa Nova

(left) and Bossa Nova (right). As can be seen in both histograms the prominant periodicities or candidate tempos are clearly visible. Once the tempo of the piece is identified the beat locations can be calculated by locally fitting tempo hypothesis with regularly spaced peaks of the onset strength signal. There has been a systematic evaluation of different tempo induction methods [40] in the context of the Music Information Retrieval Evaluation Exchange [25].

Frequently a subband decomposition is performed so that periodicities at different frequency ranges can be identified. For example the bass drum sound will mostly affect low frequency whereas a snare hit will affect all frequencies. Figure 5.7 shows a schematic diagram of the Beat Histogram calculation method described originally in [101]. A Discrete Wavelet Transform (DWT) filterbank is used as the front-end, followed by multiple channel envelop extraction and periodicity detection.

5.3.3 Rhythm representations

The beat histogram described in the previous section can be viewed as a song-level representation of rhythm. In addition to the tempo and related periodicities the total energy and/or height of peaks represent the amount of self-similarity that the music has. This quantity has been termed beat strength and has been shown to be a perceptually salient characteristic of rhythm. For example a piece of rap music with tempo of 80BPM will have more beat strength than a piece of country music at the same tempo. The spread of energy around each peak indicates the amount of tempo variation and the ratios between the tempos of the prominent peaks give

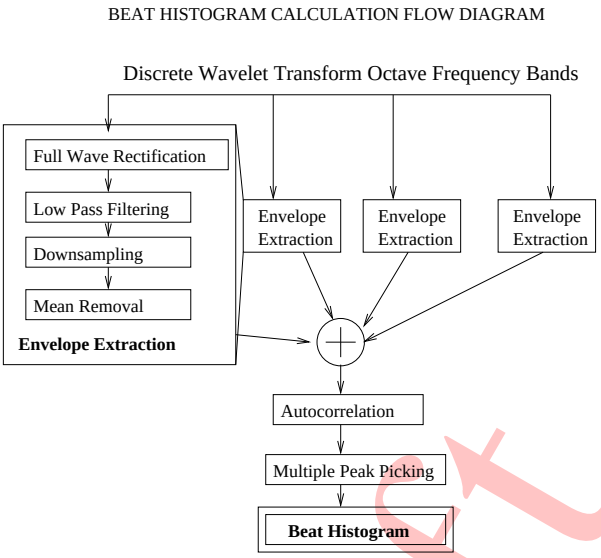


Figure 5.7: *Beat Histogram Calculation.*

hints about the time signature of the piece. A typical approach is to further reduce the dimensionality of the Beat Histogram by extracting characteristic features such as the location of the highest peak and its corresponding height [101]. A thorough investigation of various features for characterizing rhythm can be found in Gouyon [39].

An alternative method of computing a very similar representation to the Beat Histogram is based on the self-similarity matrix and termed the Beat Spectrum [33]. Another approach models the rhythm characteristics of patterns as a sequence of audio features over time [85]. A dynamic time warping algorithm can be used to align the time axis of the two sequences and allow their comparison. Another more recent approach is to identify rhythmic patterns that are characteristic of a particular genre automatically [96] and then use their occurence histogram as a feature vector.

One interesting question is whether rhythm representations should be tempo invariant or variant. To some extent the answer depends on the task. For example if one is trying to automatically classify speed metal (a genre of rock music) pieces then absolute tempo is a pretty good feature to include. On the other hand classifying something as a Waltz has more to do with the ratios of periodicities

rather than absolute tempo. Representations that are to some degree tempo invariant have also been explored. Dynamic periodicity wrapping is a dynamic programming technique used to align average periodicity spectra obtained from the onset strength signal [47]. Tempo invariance is achieved through the alignment process. The Fast Melin Transform is a transform that is invariant to scale. It has been used to provide a theoretically tempo invariant (under the assumption that tempo is scaled uniformly throughout the piece) representation by taking the FMT of the autocorrelation of the onset strength function [48]. An exponential grouping of the lags of the autocorrelation function of the onset strength signal can also be used for a tempo invariant representation [51]. Beat Histograms can also be used as the basis for a tempo invariant representation by using a logarithmically-spaced lag-axis [42]. The algorithm requires the estimation of a reference point. Experiments comparing four periodicity representations in the spectral or temporal domains using the autocorrelation and the discrete fourier transform (DFT) of the onset strength signal have been also conducted [86].

5.4 Pitch/Harmony Features

In other cases, for example in cover song identification or automatic chord detection, it is desired to have a representation that is related to the pitch content of the music rather than the specifics of the instruments and voices that are playing. Conceivably a fully automatically transcribed music score could be used for this purpose. Unfortunately current automatic transcription technology is not robust enough to be used reliably.

Instead the most common pitch content representation is the Pitch and Pitch Class Profile (PCP) (other alternative names used in literature are pitch histograms and chroma vectors). The pitch profile measures the occurrence of specific discrete musical pitches in a music segment and the pitch class profile considers all octaves equivalent essentially folding the pitch profile into 12 pitch classes. The pitch profile and pitch class profile are strongly related to the underlying harmony of the music piece. For example, a music segment in C major is expected to have many occurrences of the discrete pitch classes C, E, and G that form the C major chord. These representations are used for a variety of tasks including automatic key identification [93, 65], chord detection [93, 65], cover song detection [29, 92, 69], and polyphonic audio-score alignment [81].

There are two major approaches to the computation of the PCP. The first approach directly calculates the PCP by appropriately mapping and folding all the magnitude bins of a Fast Fourier Transform. The terms chroma and chromagram are used to describe this process [7]. Each FFT bin is mapped to its closest note, according to:

$$f(p) = 440 * 2^{p-69/12} \tag{5.9}$$

where p is the note number. This is equivalent to segmenting the magnitude spectrum into note regions. The average magnitude within each region is calculated resulting in a pitch histogram representation. The pitch histogram is then folded so that all octaves map to the same pitch class resulting into a vector of size 12. The FFT-based calculation of chroma has the advantage that it is efficient to compute and has consistent behavior throughout the song. However, it is affected by non-pitched sound sources such as drums, and the harmonics of pitch sound sources are mapped to pitches which reduces the potential precision of the represenation.

An alternative is to utilize multiple pitch detection to calculate the pitch and pitch class profiles. If the multiple pitch detection was perfect then this approach would eliminate the problems of chroma calculation, however in practice pitch detection especially in polyphonic audio is not particularly robust. The errors tend to average out but there is no consensus about whether the pitch-based or FFT-based approach is better. Multiple pitch detection typically operates in an incremental fashion. Initially the dominant pitch is detected and the frequencies corresponding to the fundamental and the associated harmonics are removed for example by using comb filtering. Then the process is repeated to detect the second most dominant pitch.

Pitch detection has been a topic of active research for a long time mainly due to its important in speech processing. In this chapter we briefly outline two common approaches. The first approach is to utilize time-domain autocorrelation of the signal [12]:

$$R(\tau) = \frac{1}{N} \sum_{n=0}^{N-1-m} x[n]x[n+m] \quad 0 \leq m < M \tag{5.10}$$

An alternative used in the YIN pitch extraction method [23] is based on the difference function:

$$dt = \sum_{n=0}^{N-1} (x[n] - x[n+\tau])^2 \tag{5.11}$$

The dips in the difference function correspond to periodicities. In order to reduce the occurrence of subharmonic errors, YIN employs a cumulative mean function which de-emphasizes higher period dips in the difference function.

Independently of their method of calculation PCPs can be calculated using fixed size analaysis window. This occassionally will result in inaccuracies when a window contains information from two chords. The use of beat synchronous features [29] can help improve the results as by considering windows that are aligned with beat locations it is more likely the the pitch content information remains stable within an analysis window.

5.5 Other Audio Features

In addition to timbral texture, rhythm and pitch content there are other facets of musical content that have been used as the basis for audio feature extraction. Another important source of information is the instrumentation of audio recordings i.e what instruments are playing at any particular time of an audio recording. For example it is more likely that a saxophone will be part of a jazz recording than a recording of country music although of course there are exceptions. The classification of musical instrument in a polyphonic context has been explored [91] but so far has not been evaluated in the context of other music data mining tasks such as classification.

So far all the features described characterize the mixture of sound sources that comprise the musical signal. Another possibility is to characterize individually each sound source. Feature extraction techniques for certain types of sound sources have been proposed but they are not widely used partly due to the difficulty of separating and/or characterizing individual sound sources in a complex mixture of sounds such as music. An example is automatic singer identification. The most basic approach is to use features developed for voice/speech identification directly on the mixed audio signal [59]. More sophisticated approaches first attempt to identify the parts of the signal containing vocals and in some cases even attempt to separately characterize the singing voice and reduce the effect of accompaniment [34]. Another type of sound source that has been explored for audio feature extraction is the bass line [60, 97].

In many modern pop and rock recordings each instrument is recorded separately and the final mix is created by a recording producer/engineer(s) who among other transformations add effects such as reverb or filtering and spatialize in-

dividual tracks using stereo panning cues. For example the amount of stereo panning and placement of sources remains constant in older recordings that tried to reproduce live performances compared to more recent recordings that would be almost impossible to realize in a live setting. Stereo panning features have recently been used for audio classiffication [102, 105].

5.6 Bag of codewords

5.7 Spectral Features

Spectral Centroid and spectral spread: These features are computed on the power spectrum of the sound. They are respectively the mean and standard deviation of the frequencies in the power spectrum weighted by the power index.

5.8 Mel-Frequency Cepstral Coefficients

5.9 Pitch and Chroma

5.10 Beat, Tempo and Rhythm

5.11 Modeling temporal evolution and dynamics

5.12 Pattern representations

5.13 Stereo and other production features

Chapter 6

Context Feature Extraction

6.1 Extracting Context Information About Music

In addition to information extracted by analyzing the audio content of music there is a wealth of information that can be extracted by analyzing information on the web as well as patterns of downloads/listening. We use the general term musical context to describe this type of information. In some cases such as song lyrics the desired information is explicitly available somewhere on the web and the challenge is to appropriately filter out irrelevant information from the corresponding web-pages. Text-based search engines such as Google and Bing can be leveraged for the initial retrieval that can then be followed by some post-processing based on heuristics that are specific to the music domain. Other types of information are not as straightforward and can require more sophisticated mechanisms such as term weighting used in text retrieval systems or natural language processing techniques such as entity detection. Such techniques are covered in detail in the literature as they are part of modern day search engines. As an illustrative example we will consider the problem of detecting the country of origin of a particular artist. As a first attempt one can query a search for various pairs of artist name and countries and simply count the number of pages returned. The country with the highest number of pages returned is returned as the country of origin. A more sophisticated approach is to analyze the retrieved web pages using term weighting. More specifically consider country c as a term. The document frequency $DF(c, a)$

is defined as the total number of web pages retrieved for artist a in which the country term c appears at least once. The term frequency $TF(c, a)$ is defined as the total number of occurrences of the country term c in all pages retrieved for artist a. The basic idea of term frequency-inverse document frequency weighting (TF-IDF) is to "penalize" terms that appear in many documents (in our case the documents retrieved for all artists) and increase the weight of terms that occur frequently in the set of web pages retrieved for a specific artists. There are several TF-IDF weighting schemes. For example a logarithmic formulation is:

$$TFIDF(c, a) = ln(1 + TF(c, a)) * ln\left(1 + \frac{n}{DF(c)}\right) \tag{6.1}$$

where DF(c) is the document frequency of a particular country c over the documents returned for all artists a. Using the above equation the weight of every country c can be calculated for a particular artist query a. The country with the highest weight is then selected as the the predicted country of origin.

Chapter 7

Analysis

Essentially, all models are wrong but some are useful.
George Box

In the previous chapters we have described how music in digital audio formats can be represented using time-frequency representations as well as how audio features at different time scales can be extracted based on these representations. In this chapter we review the techniques used to analyze these audio features and extract musical content information from them. Most of these techniques originate in the related fields of machine learning, pattern recognition and data mining. Going further back most of them are based on ideas from statistics and probability theory. Computers are particularly suited to solving tasks that are well-defined and for which there is a clear sequence of steps that guarantees finding a solution. For example sorting a list of numbers or calculating compounded interest rates are the types of tasks that computers excel at and perform daily. A solution to such a problem can be easily verified and is fixed for a particular set of input data.

However, there are many problems that we are interested in solving for which there is no known sequence of steps that allows us to solve them and for which there is no perfect well defined solution. A simple example would be predicting the weather in a particular location based on measurements of humidity, pressure, and temperature from various weather stations in the surrounding area. If someone is given this problem it is unclear what procedure should be followed and although an algorithm that would correctly predict the weather with 100% accuracy would be ideal, any algorithm that performs significantly better than random guessing

would still be useful. The traditional approach to solving this problem, before the widespread use of computers and advances in machine learning and data mining, was to plot the measurements over several days, try to fit some simple mathematical functions, and try to come up with some prediction rules expressed as an algorithm that can be executed on a computer. Such an approach requires significant effort and as the number of measurements to be considered increases it becomes very difficult if not impossible to utilize all the available information. One key observation is that for many such problems although there is no well defined optimal solution and no clear procedure for obtaining it, it is possible to compare two different algorithms and decide which one performs better i.e there is a way to evaluate the effectiveness of an algorithm. For example in the weather prediction example we can execute the algorithm over historical data for which the weather that should be predicted is known and evaluate how many times the different algorithms under consideration make correct predictions on this data.

As defined by Mitchell [**?**] a machine learning algorithm is a computer program P that tries to solve a particular task T and whose performance on this task as measured by some metric E improves over the course of its execution. For example, a computer chess program "learns" from playing chess (T) with human players (E) as measured by the number of games it wins (P), if the number of games it wins (P) increases as it plays more games.

A common way this improvement with experience is achieved is by considering more data. Returning to the weather prediction example, in a machine learning approach the acquired measurements as well as the associated "ground truth" would be the input to the algorithm. The algorithm would somehow model the association between the measurements and the desired output without requiring any decision to be made by the programmer and once "trained" could be applied to new data. The ability of the algorithm to predict the weather would improve based on the availability of historical data. Using a machine learning/data mining approach large amounts of data can be utilized and no explicit programming of how the problem is solved needs to be done. This process might seem a little bit strange and abstract at this point but hopefully will become clearer through concrete examples in the following sections. The techniques described in this chapter are used in several MIR tasks described in the second part of this book.

7.1 Feature Matrix, Train Set, Test Set

In many analysis and mining tasks the data of interest is represented as vectors of numbers. By having each object of interest represent as a finite vector of numbers (features) the domain specific information is encapsulated in the feature and generic machine learning algorithms can be developed. For example, the same classification algorithm with different features as input can be used to perform face recognition (with features based on image analysis), music genre classifiaction (with features based on audio analysis), or document categorization (with features based on text analysis).

A feature matrix is a 2D matrix where each row (also known as instance or sample) is a feature vector consisting of numbers (also known as attributes or features) characterizing a particular object of interest. Formally it can be notated as:

$$\mathbf{X} = \mathbf{x_i} \in \mathbf{R^d}, i = 1, \ldots, n \tag{7.1}$$

where i is the instance index, n is the number of instances, and d is the dimensionality of the feature vector.

In supervised learning the feature matrix is associated with a vector of ground truth classification labels typically represented as integers for each instance y_i. The training set consists of the feature matrix and associated labels:

$$\mathbf{T} = (\mathbf{X}, \mathbf{y}) = (\mathbf{x_i}, \mathbf{y_i}). \tag{7.2}$$

7.2 Similarity and distance metrics

Euclidean, Mahalanobis, Earth Mover, Kullback-Leibler

7.3 Classification

Classification is one of the most common problems in pattern recognition and machine learning. The basic idea is straightforward to describe. The goal of a classification system is to "classify" objects by assigning to them a unique label from a finite, discrete set of labels (also known as classes or categories).

In classic supervised learning, a "training" set of feature vectors representing different objects is provided together with associated "ground truth" class labels. After the classifier is "trained" it can then be used to "predict" the class label of objects that it has not encountered before using the feature vectors corresponding to these objects.

A classifier is an algorithm that operates in two phases. In the first phase called training, it takes as input the training set T and analyzes it in order to compute what is called a model. The model is some kind of representation, typically a vector of parameters, that characterizes the particular classification problem for a given type of classifier. In the second phase, called prediction or testing, the trained model is used to predict (estimate) the classification labels for a new feature matrix called the testing set.

7.3.1 Evaluation

Before describing different classification algorithms we discuss how their effectiveness can be evaluated. Having a good understanding of evaluation is even more important than understanding how different algorithms work as in most cases the algorithm can be simply treated as a black box.

The goal of a classifier is to be able to classify objects it has not encountered before. Therefore in order to get a better estimate of its performance on unknown data it is necessary to use some of the instances labeled with ground truth for testing purposes and not take them into account when training. The most common such evaluation scheme is called K-fold cross-validation. In this scheme the set of labeled instances is divided into K distinct subsets (folds) of approximately equal sizes. Each fold is used for testing once with the $K-1$ remaining folds used for training. As an example if there are 100 feature vectors then each fold will contain 10 feature vectors with each one of them being used one time for testing and $K-1$ times for training. The most common evaluation metric is classification accuracy which is defined as the percentage of testing feature vectors that were classified correctly based on the ground truth labels. When classifying music tracks a common post-processing technique that is applied is the so-called artist filter which ensures that the feature vectors corresponding to tracks from the same artist are not split between training and testing and are exclusively allocated only to one of them. The rationale behind artist filtering is that feature vectors from the same artist will tend to be artificially related or correlated due to similarities in the recording process and instrumentation. Such feature vectors will be classified

more easily if included in both training and testing and maybe inflate the classification accuracy. Similar considerations apply to feature vectors from the same music track if each track is represented by more than one in which case a similar track filter should be applied.

XXX Add explanations of bootstrapping, leave-one-out, percentage split XXX stratification

The most common evaluation metric for automatic classification is accuracy which is simply defined as the number of correctly classified instances in the testing data. It is typically expressed as a percentage. Additional insight can be provided by examining the confusion matrix which is a matrix that shows the correct classifications in the diagonal and shows how the misclassification are distributed among the other class labels.

XXX Figure with hypothetical example XXX Figure with confusion matrix

7.3.2 Generative Approaches

naive bayes, gmm

7.3.3 Discriminative

perceptron, svm, ann, decision tree perceptron, svm, decision tree, adaboost, KNN

7.3.4 Ensembles

adaboost

7.3.5 Variants

multi-label, hierarchical, co-clustering, online, multi-instance,

7.4 Clustering

k-means, spectral clustering, hierarchical

7.5 Dimensionality Reduction

7.5.1 Principal Component Analysis

Principal component analysis (PCA) coverts a set of feature vectors with possibly correlated attributes into a set of feature vectors where the attributes are linearly uncorrelated. These new transformed attributes are called the principal components and when the method is used for dimensionality reduction their number is less than the number of original attributes. Intuitively this transformation can be understood as a projection of the original feature vectors to a new set of orthogonal axes (the principal components) such that each succeeding axis explains the highest variance of the original data-set possible with the constraint that it is orthogonal to the preceding component. In a typical application scenario where each song is represented by a 70 dimensional feature vector, the application of PCA to these feature vectors could be used to convert them to 3 dimensional feature vectors which can be visualized as points in a 3D space. A common way of calculating PCA is based on the covariance matrix which defined as:

$$C = \frac{1}{N} \sum B \times B^* \tag{7.3}$$

where $*$ denotes the transpose operator and B is the matrix resulting from subtracting the empirical mean of each dimension of the original data matrix consisting of the N feature vectors. The eigenvectors and eigenvalues of this covariance matrix are then computed and sorted in order of decreasing eigenvalue and the first Km, where K is the desired number of reduced dimensions, are selected as the new basis vectors. The original data can then be projected into the new space spanned by these basis functions i.e the principal components. PCA is a standard operation in statistics and it is commonly available in software packages dealing with matrices.

One of the potential issues with PCA for music visualization is that because it tries to preserve as much as possible the distances between points from the original feature space to the transformed feature space it might leave large areas of the available space empty of points. This is particularly undesirable in interfaces based on various types of touch surfaces where in the ideal case everywhere a user might press should trigger some music. Self-organizing maps are an approach that attempts to perform both dimensionality reduction and clustering at the same time while mostly preserving topology but not distances. It can result in more

dense transformed feature spaces that are also discrete in nature in contrast to PCA which produces a transformed continuous space that needs to be discretized.

7.5.2 Self-Organizing Maps

The traditional SOM consists of a 2D grid of neural nodes each containing an n-dimensional vector, $\mathbf{x}(\mathbf{t})$ of data. The goal of learning in the SOM is to cause different neighboring parts of the network to respond similarly to certain input patterns. This is partly motivated by how visual, auditory and other sensory information is handled in separate parts of the cerebral cortex in the human brain. The network must be fed a large number of example vectors that represent, as closely as possible, the kinds of vectors expected during mapping. The data associated with each node is initialized to small random values before training. During training, a series of n-dimensional vectors of sample data are added to the map. The "winning" node of the map known as the *best matching unit* (BMU) is found by computing the distance between the added training vector and each of the nodes in the SOM. This distance is calculated according to some pre-defined distance metric which in our case is the standard Euclidean distance on the normalized feature vectors.

Once the winning node has been defined, it and its surrounding nodes reorganize their vector data to more closely resemble the added training sample. The training utilizes competitive learning. The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU. The time-varying learning rate and neighborhood function allow the SOM to gradually converge and form clusters at different granularities. Once a SOM has been trained, data may be added to the map simply by locating the node whose data is most similar to that of the presented sample, *i.e.* the winner. The reorganization phase is omitted when the SOM is not in the training mode.

The update formula for a neuron with representative vector N(t) can be written as follows:

$$\mathbf{N}(t+1) = \mathbf{N}(t) + \Theta(v,t)\alpha(t)(\mathbf{x}(t) - \mathbf{N}(t)) \tag{7.4}$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient and $x(t)$ is the input vector. The neighborhood function $\Theta(v,t)$ depends on the lattice distance between the BMU and neuron v.

PCA, LDA, Self-Organizing Maps, FastMap,

7.6 Modeling Time Evolution

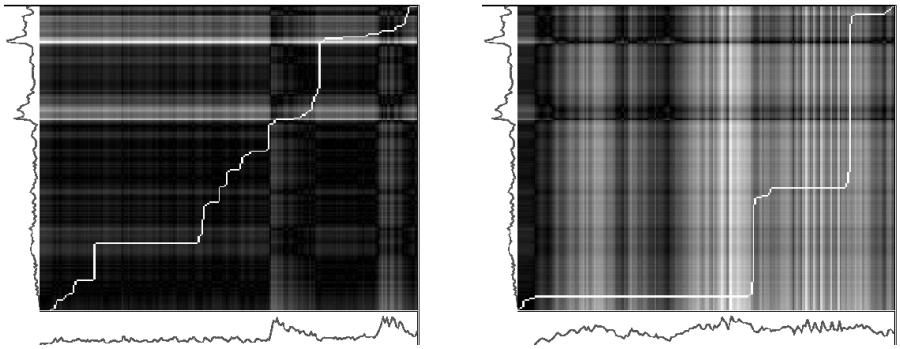
7.6.1 Dynamic Time Warping

More formally the problem is, given two sequences of feature vectors with different lengths and timings, find the optimal way of "elastically" transforming by the sequences so that they match each other. A common technique used to solve this problem and also frequently employed in the literature for cover song detection is Dynamic Time Warping (DTW) a specific variant of dynamic programming. Given two time series of feature vectors $X = (x_1, x_2, \ldots, x_M)$ and $Y = (y_1, y_2, \ldots, y_N)$ with $X, Y \in \mathbb{R}^d$ the DTW algorithm yields an optimal solution in $O(MN)$ time where M and N are the lengths of the two sequences. It requires a local distance measure that can be used to compare individual feature vectors which should have small values when the vectors are similar and large values when they are different:

$$d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \geq 0 \tag{7.5}$$

The algorithm starts by building the distance matrix $C \in \mathbb{R}^{M \times N}$ representing all the pairwise distances between the feature vectors of the two sequences. The goal of the algorithm is to find the **alignment or warping** path which is a correspondence between the elements of the two sequences with the boundary constraint that the first and last elements of the two sequences are assigned to each other. Intuitively for matching sequences the alignment path will be roughly diagonal and will run through the low-cast areas of the cost matrix. More formally the alignment is a sequence of points $(p_i, p_j) \in [1 : M] \times [1 : N]$ for which the starting and ending points must be the first and last points of the aligned sequences, the points are time-ordered and each step size is constrained to either move horizontally, vertically or diagonally. The cost function of the alignment path is the sum of all the pairwise distances associated with its points and the alignment path that has the minimal cost is called the **optimal alignment path** and is the output of the DTW.

Figure 14.1 shows two distance matrices that are calculated based on energy contours of different orchestra music movements. The left matrix is between two performances by different orchestras of the same piece. Even though the timing and duration of each performance is different they exhibit a similar overall energy envelope shown by the energy curves under the two axes. The optimal alignment path computed by DTW is shown imposed over the distance matrix. In contrast



(a) Good Alignment (b) Bad Alignment

Figure 7.1: *Similarity Matrix between energy contours and alignment path using Dynamic Time Warping*

the matrix on the right shows the distance matrix between two unrelated orchestral movements where it is clear there is no alignment.

7.6.2 Hidden Markov Models

7.6.3 Kalman and Particle Filtering

7.7 Further Reading

cite Mitchell, Duda and Hart, Theodoridis and Bishop.

Part II

Tasks

Chapter 8

Similarity Retrieval

Similarity retrieval (or query-by-example) is one of the most fundamental MIR tasks. It is also one of the first tasks that were explored in the literature. It was originally inspired by ideas from text information retrieval and this early influence is reflected in the naming of the field. Today most people with computers use search engines on a daily basis and are familiar with the basic idea of text information retrieval. The user submits a query consisting of some words to the search engine and the search engine returns a ranked list of web pages sorted by how relevant they are to the query.

Similarity retrieval can be viewed as an analogous process where instead of the user querying the system by providing text the query consists of an actual piece of music. The system then responds by returning a list of music pieces ranked by their similarity to the query. Typically the input to the system consists of the query music piece (using either a symbolic or audio representation) as well as additional metadata information such as the name of the song, artist, year of release etc. Each returned item typically also contains the same types of meta-data. In addition to the audio content and meta-data other types of user-generated information can also be considered such as ranking, purchase history, social relations and tags. Similarity retrieval can also be viewed as a basic form of playlist generation in which the returned results form a playlist that is "seeded" by the query. However more complex scenarios of playlist generation can be envisioned. For example a start and end seed might be specified or additional constraints such as approximate duration or minimum tempo variation can be specified. Another variation is based

on what collection/database is used for retrieval. The term playlisting is more commonly used to describe the scenario where the returned results come from the personal collection of the user, while the term recommendation is more commonly used in the case where the returned results are from a store containing a large universe of music. The purpose of the recommendation process is to entice the user to purchase more music pieces and expand their collection. Although these three terms (similarity retrieval, music recommendation, automatic playlisting) have somewhat different connotations the underlying methodology for solving them is similar so for the most part we will use them interchangeably. Another related term that is sometimes used in personalized radio in which the idea is to play music that is personalized to the preferences to the user.

One can distinguish three basic approaches to computing music similarity. Content-based similarity is performed by analyzing the actual content to extract the necessary information. Metadata approaches exploit sources of information that are external to the actual content such as relationships between artists, styles, tags or even richer sources of information such as web reviews and lyrics. Usage-based approaches track how users listen and purchase music and utilize this information for calculating similarity. Examples include collaborative filtering in which the commonalities between purchasing histories of different users are exploited, tracking peer-to-peer downloads or radio play of music pieces to evaluate their "hotness" and utilizing user generated rankings and tags.

There are trade-offs involved in all these three approaches and most likely the ideal system would be one that combines all of them intelligently. Usage-based approaches suffer from what has been termed the "cold-start" problem in which new music pieces for which there is no usage information can not be recommended. Metadata approaches suffer from the fact that metadata information is frequently noisy or inaccurate and can sometimes require significant semi-manual effort to clean up. Finally content-based methods are not yet mature enough to extract high-level information about the music.

From a data mining perspective similarity retrieval can be considered a ranking problem. Given a query music track q and a collection of music tracks D the goal of similarity retrieval is to return a ranked list of the music tracks in D sorted by similarity so that the most similar objects are at the top of the list. In most approaches this ranking is calculated by defining some similarity (or distance) metric between pairs of music tracks. The most basic formulation is to represent each music track as a single feature vector of fixed dimensionality $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and use standard distance metrics such as L1 (Manhattan) or L2 (Euclidean) or Mahalanobis on the resulting high dimensional space. This

feature vector is calculated using audio feature extraction techniques as described in the following sections. Unless the distance metric is specifically designed to handle features with different dynamic ranges the feature vectors are typically normalized for example by scaling all of them so that their maximum value over the dataset is 1 and their minimum value is 0. A more complex alternative is to treat each music track as a distribution of feature vectors. This is accomplished by assuming that the feature vectors are samples of an unknown underlying probability density function that needs to be estimated. By assuming a particular parametric form for the pdf (for example a Gaussian Mixture Model) the music track is then represented as a parameter vector θ that is estimated from the data. This way the problem of finding the similarity between music tracks is transformed to the problem of somehow finding the difference between two probability distributions. Several such measures of probability distance have been proposed such as histogram intersection, symmetric Kullback-Leibler divergence and earth mover's distance. In general computation of such probabilistic distances are more computationally intensive than geometric distances on feature vectors and in many cases require numerical approximation as they can not be obtained analytically. An alternative to audio feature extraction is to consider similarity based on text such as web pages, user tags, blogs, reviews, and song lyrics. The most common model when dealing with text is the so called "bag of words" representation in which each document is represented as an unordered set of its words without taking into account any syntax or structure. Each word is assigned a weight that indicates the importance of the word for some particular task. The document can then be represented as a feature vector comprised of the weights corresponding to all the words of interest. From a data mining perspective the resulting feature vector is no different than the ones extracted from audio feature extraction and can be handled using similar techniques. In the previous section we described a particular example of text based feature extraction for the purpose of predicting the country of origin as an artist.

As an example of how a text-based approach can be used to calculate similarity consider the problem of finding how similar two artists A and B are. Each artist is characterized by a feature vector consisting of term weights for the terms they have in common. The cosine similarity between the two feature vectors is defined as the cosine of the angle between the vectors and has the property that it is not affected by the magnitude of the vector (which would correspond to the absolute number of times terms appear and could be influenced by the popularity of the artist):

$$sim(A,B) = cos\theta = \frac{\sum_t A(t) \times B(t)}{\sqrt{\sum_t A(t)^2} \times \sqrt{\sum_t B(t)^2}} \tag{8.1}$$

Another approach to calculating similarity is to assume that the occurrence of two music tracks or artists within the same context indicates some kind of similarity. The context can be web pages (or page counts returned by a search engine), playlists, purchase histories, and usage patterns in peer-to-peer (P2P) networks. Collaborative filtering (CF) refers to a set of techniques that make recommendations to users based on preference information from many users. The most common variant is to assume that the purchase history of a particular user (or to some extent equivalently their personal music collections) is characteristic of their taste in music. As an example of how co-occurrence can be used to calculate similarity, a search engine can be queried individually for documents that contain a particular artist A and B, as well as documents that contain both A and B. The artist similarity between A and B can then be found by:

$$sim(A,B) = \frac{co(A,B)}{min(co(A), co(B))} \tag{8.2}$$

where $co(X)$ is the number of pages returned for query X or the co-occurrences of A and B in some context. A similar measure can be defined based co-occurrences between tracks and artists in playlists and compilation albums based on conditional probabilities:

$$sim(A,B) = \frac{1}{2} * \left(\frac{co(A,B)}{co(A)} + \frac{co(A,B)}{co(B)} \right) \tag{8.3}$$

Co-occurrences can also be defined in the context of peer-to-peer networks by considering the number of users that have both artists A and B in their shared collection. The popularity bias refers to the problem of popular artists appearing more similar than they should be due to them occurring in many contexts. A similarity measure can be designed to downweight the similarity between artists if one of them is very popular and the other is not (the right-hand part of the following equation):

$$sim(A,B) = \frac{C(A,B)}{C(B)} * \left(1 - \frac{|C(A) - C(B)|}{C(Max)} \right) \tag{8.4}$$

where $C(Max)$ is the number of times the most popular artist appears in a context.

8.0.1 Evaluation of similarity retrieval

One of the challenges in content-based similarity retrieval is evaluation. In evaluating mining systems ideally one can obtain ground truth information that is identical to the outcome of the mining algorithm. Unfortunately this is not the case in similarity as it would require manually sorting large collections of music in order of similarity to a large number of queries. Even for small collections and not many queries collecting such data would be extremely time consuming and practically impossible. Instead the more common approach is to only consider the top K results for each query where K is a small number and have users annotate each result as relevant or not relevant. Sometimes a numerical discrete score is used instead of a binary relevance decision. Another possibility that has been used is to assume that tracks by the same artist or same genre should be similar and use such groupings to assign relevance values.

Evaluation metrics based on information retrieval can be used to evaluate the retrieved results for a particular query. They assume that each of the returned results has a binary annotation indicating whether or not it is relevant to the query. The most common one is the *F-measure* which is a combination of the simpler measures of *Precision* and *Recall*. *Precision* is the fraction of retrieved instances that are relevant. *Recall* is the fraction of relevant instances that are retrieved. As an example consider a music similarity search in which relevance is defined by genre. If for a given query of Reggae music it returns 20 songs and 10 of them are also Reggae the precision for that query is $\frac{10}{20} = 0.5$. If there are a total of 30 reggae songs in the collection searched then the *Relevance* for that query is $\frac{10}{30} = 0.33$. The *F-measure* is defined as the harmonic mean of *Precision* P and *Recall* R.

$$F = 2 \times \frac{P \times R}{P + R} \tag{8.5}$$

These three measures are based on the list of documents returned by the system without taking into account the order they are returned. For similarity retrieval a more accurate measure is the *Average Precision* which is calculated by computing the precision and recall at every position in the ranked sequence of documents, creating a precision-recall curve and computing the average. This is equivalent to the following finite sum:

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{\#relevantdocuments} \tag{8.6}$$

where P(k) is the precision at list position k and $rel(k)$ is an indicator function that is 1 if the item at list position (or rank) k is a relevant document and 0 otherwise.

All of the measures described above are defined for a single query. They can easily be extended to multiple queries by taking their average across the queries. The most common way of evaluating similarity systems with binary relevance ground-truth is the *Mean Average Precision* (MAP) which is defined as the mean of the *Average Precision* across a set of queries.

The Music Information Retrieval Evaluation Exchange (MIREX) is an annual evaluation benchmark in which different groups submit algorithms to solve various MIR tasks and their performance is evaluated using a variety of subjective and objective metrics. Table **??** shows representative results of the music similarity and retrieval task from 2010. It is based on a dataset of 7000 30-second audio clips drawn from 10 genres. The objective statistics are the precision at 5, 10, 20 and 50 retrieved items without counting entries by the artist (artist filtering). The subjective statics are based on human evaluation of approximately 120 randomly selected queries and 5 results per query. Each result is graded with a fine score (between 0 and 100 with 100 being most similar) and a broad score (0 not similar, 1 somewhat similar, 2 similar) and the results are averaged. As can be seen all automatic music similarity systems perform significantly better than the random baseline (RND). They differ in terms of the type of extracted features utilized, the decision fusion strategy (such as simple concatenation of the different feature sets or empirical combinations of distances from the individual feature sets), and whether post-processing is applied to the resulting similarity matrix. There is also a strong correlation between the subjective and objective measures although it is not perfect (for example SSPK2 is better than PSS1 in terms of subjective measures but worst in terms of objective measures).

	FS	BS	P@5	P@10	P@20	P@50
RND	17	0.2	8	9	9	9
TLN3	47	0.97	48	47	45	42
TLN2	47	0.97	48	47	45	42
TLN1	46	0.94	47	45	43	40
BWL1	50	1.08	53	51	50	47
PS1	55	1.22	59	57	55	51
PSS1	55	1.21	62	60	58	55
SSPK2	57	1.24	59	58	56	53

Table 8.1: 20120 MIREX Music Similarity and Retrieval Results

Chapter 9

Classification and Tagging

9.1 Introduction

The combination of audio feature extraction extraction followed by a stage of machine learning forms the basis of a variety of different MIR classification systems that have been proposed. Probably the most widely studied problem is automatic genre or style classification. More recently emotion and mood recognition have also been tackled. The task of identifying musical instruments either in monophonic or polyphonic recordings is something listers, especially if they are musically trained, can perform with reasonable accuracy. Instrument recognition is another well explored MIR classification topic covered in this chapter.

9.2 Genre Classification

In the last 15 years musical genre classification of audio signals has been widely studied and can be viewed as a canonical problem in which audio feature extraction followed by supervised learning has been used. Frequently new approaches to audio feature extraction are evaluated in the context of musical genre classification. Examples of such new feature sets include sparse overcomplete repre-

sentations [28] as well as bio-inspired joint acoustic and modulation frequency represenations [84].

Audio classification had a long history in the area of Speech Recognition but it was only more recently applied to music starting around 2000. Even though there was some earlier work [62, 94] a good starting point for audio-based musical genre classification is the system that was proposed in 2002 by Tzanetakis [101]. This work was influential as it was the first time that a relatively large, at least for the time, audio collection was used (1000 clips each 30 seconds long evenly distributed in 10 genres). Another important contribution was the use of features related to pitch and rhythmic content that were specific to music. Probably most importantly autoatic genre classification was a problem that clearly combined digital signal processing and classic machine learning with an easy to explain evaluation methodology that did not require user involvement. In the next 15 years, a variety of automatic genre classsification systems were proposed, frequently utilizing the infamous GTZAN dataset for evaluation. There have also been valid criticisms about overuse of this dataset as well as methodological issues with automatic genre classification.

There is a large body of work in musical genre classification (the term recognition is also frequently used). One of the most thorough and complete bibliographies on this topic can be found in the thorough survey paper of evaluation in musical genre recognition by Bob Sturm [] who cites almost 500 publications related to this topic and provides a lot of information about the associated history, different ways of evaluation, and the datasets that have been used. There have also been several survey of musical genre recognition [].

9.2.1 Formulation

We begin our exposition by describing more formally the problem of automatic genre classification in its most common and basic form.

Once the audio features are extracted they need to be used to "train" a classifier using supervised learning techniques. This is accomplished using all the labeled audio feature representations for a training collection of tracks. If the audio feature representation has been summarized over the entire track to a single high-dimensional feature vector then this corresponds to the "classic" formulation of classification and any machine learning classifier can be used directly. Examples of classifiers that have been used in the context of audio-based music classification

include: Gaussian Mixture Models [4, 101], Support Vector Machines [75, 66], and AdaBoost [10].

Another alternative is to perform classification in smaller segments and then aggregate the results using majority voting. A more complicated approach (frequently called bag-of-frames) consists of modeling each track using distribution estimation methods, for example a Gaussian Mixture Model trained using the EM-algorithm. In this case each track is representanted as a probability distribution rather than a single high-dimensional point (feature vector). The distance between the probability distributions can be estimated for example using KL-divergence or approximations of it for example using the Monte-Carlo method depending on the particular parametric form used for density estimation. By establishing a distance metric between tracks it is possible to perform retrieval and classification by simple techniques such as k-nearest neighbors [4].

9.2.2 Evaluation

Evaluation of classification is relatively straightforward and in most ways identical to any classification task. The standard approach is to compare the predicted labels with ground truth labels. Common metrics include classification accuracy as well as retrieval metrics such as precision, recall, f-measure. When retrieval metrics are used it is assumed that for a particular query revelant documents are the tracks with the same genre label. Cross-validation is a technique frequently used in evaluating classification where the labeled data is split into training and testing sets in different ways to ensure that the metrics are not influenced by the particular choice of training and testing data. One detail that needs to be taken into account is the so-called album effect in which classification accuracy improves when tracks from the same album are included in both training and testing data. The cause is recording production effects that are common between tracks in the same album. The typical approach is to ensure that when performing cross-validation, tracks from the same album or artist only go to either the training or testing dataset.

Classification accuracy on the same dataset and using the same cross-validation approach can be used for comparing the relative performance of different algorithms and design choices. Interpreting the classification accuracy in absolute terms is trickier because of the subjective nature of genre labeling as has already been discussed in the section on ground truth labeling. In the early years of research in audio-based musical genre classification each research group utilized different datasets, cross-validation approaches and metrics making it hard to draw

Genre Classification	66.41
Genre Classification (Latin)	65.17
Audio Mood Classification	58.2
Artist Identification	47.65
Classical Composer Identification	53.25
Audio Tag Classification	0.28

Table 9.1: Audio-based classification tasks for music signals (MIREX 2009)

any conclusions about the merits of different approaches. Sharing datasets is harder due to copyright restrictions. The Music Information Retrieval Evaluation Exchange (MIREX) [25] is an annual event where different MIR algorithms are evaluated using a variety of metrics on different tasks including several audio-based classification tasks. The participants submit their algorithms and do not have access to the data which addresses both the copyright problem and the issue of over-fitting to a particular dataset. Table 9.1 shows representative results of of the best performing system in different audio classification tasks from MIREX 2009. With the exception of Audio Tag Classification all the results are percentages of classification accuracy. For audio tag classification the average f-measure is used instead.

9.2.3 Criticisms

9.3 Symbolic genre classification

9.4 Emotion/Mood

9.5 Instrument

[45]

9.6 Other

music/speech

9.7 Symbolic

[57]

9.8 Tagging

9.9 Further Reading

[99]

Chapter 10

Structure

[32]

10.1 Segmentation

[50]

10.2 Alignment

[?] [?]

10.3 Structure Analysis

Chapter 11

Transcription

11.1 Monophonic Pitch Tracking

[9], [58]

11.2 Transcription

[8]

11.3 Chord Detection

[6]

Chapter 12

Symbolic Music Information Retrieval

Chapter 13

Queries

13.1 Query by example

13.2 Query by humming

[77] [11] [82]

Chapter 14

Fingerprinting and Cover Song Detection

14.1 Fingerprinting

[2] [44]

14.2 Cover song detection

In addition to content-based similarity there are two related music mining problems. The goal of audio fingerprinting is to identify whether a music track is one of the recordings in a set of reference tracks. The problem is trivial if the two files are byte identical but can be considerably more challenging when various types of distortion need to be taken into account. The most common distortion is perceptual audio compression (such as mp3 files) which can result in significant alterations to the signal spectrum. Although these alterations are not directly perceptible by humans they make the task of computer identification harder. Another common application scenario is music matching/audio fingerprinting for mobile applications in which the query signal is acquired through a low quality

microphone on a mobile phone and contains significant amount of background noise and interference. At the same time the underlying signal is the same exact music recording which can help find landmark features and representations that are invariant to these distortions. Cover song detection is the more subtle problem of finding versions of the same song possibly performed by different artists or with different instrumentation and tempo. As the underlying signals are completely different it requires the use of more high level representations such as chroma vectors that capture information about the chords and the melody of the song without being affected by timbral information. In addition it requires sophisticated sequence matching approaches such as dynamic time warping (DTW) or Hidden Markov Models (HMM) to deal with the potential variations in tempo. Although both of these problems can be viewed as content-based similarity retrieval problems with an appropriately defined notion of similarity they have some unique characteristics. Unlike the more classic similarity retrieval in which we expect the returned results to gradually become less similar, in audio fingerprinting and cover song detection there is a sharper cutoff defining what is correct or not. In the ideal case copies or cover versions of the same song should receive a very high similarity score and everything else a very low similarity score. This specificity is the reason why typically approaches that take into account the temporal evolution of features are more common.

Audio fingerprinting is a mature field with several systems being actively used in industry. As a representative example we describe a landmark-based audio fingerprinting system based on the ideas used by Shazam which is a music matching service for mobile phones. In this scheme each audio track is represented by the location in time and frequency of prominent peaks of the spectrogram. Even though the actual amplitude of these peaks might vary due to noise and audio compression their actual location in the time frequency plane is preserved quite well in the presence of noise and distortion. The landmarks are combined into pairs and each pair is characterized by three numbers f_1, f_2, t which are the frequency of the first peak, the frequency of the second peak and the time between them. Both reference tracks and the query track are converted into this landmark representation. The triplets characterizing each pair are quantized with the basic idea being that if the query and a reference track have a common landmarks with consistent timing they are a match. The main challenge in an industrial strength implementation is deciding on the number of landmarks per second and the thresholds used for matching. The lookup of the query landmarks into the large pool of reference landmarks can be performed very efficiently using hashing

techniques to effectively create an inverted index which maps landmarks to the files they originate.

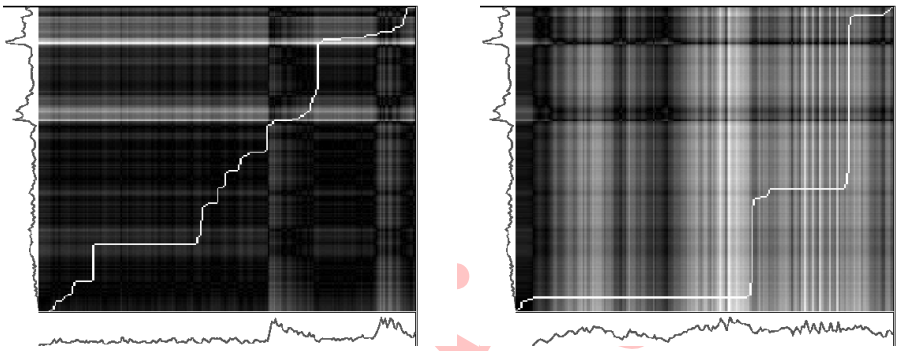
To solve the audio cover song detection problem there are two issues that need to be addressed. The first issue is to compute a representation of the audio signal that is not affected significantly by the timbre of the instruments playing but still captures information about the melody and harmony (the combination of discrete pitches that are simultaneously sounding) of the song. The most common representation used in music mining for this purpose are chroma vectors (or pitch class profiles) which can be thought of as histograms showing the distribution of energy among different discrete pitches. The second issue that needs to be addressed is how to match two sequence of feature vectors (chroma vectors in this case) that have different timing and length as there is no guarantee that a cover song is played at the same tempo as the original and there might be multiple sections with different timing each.

More formally the problem is given two sequences of feature vectors with different lengths and timings find the optimal way of "elastically" transforming by the sequences so that they match each other. A common technique used to solve this problem and also frequently employed in the literature for cover song detection is Dynamic Time Warping (DTW) a specific variant of dynamic programming. Given two time series of feature vectors $X = (x_1, x_2, \ldots, x_M)$ and $Y = (y_1, y_2, \ldots, y_N)$ with $X, Y \in \mathbb{R}^d$ the DTW algorithm yields an optimal solution in $O(MN)$ time where M and N are the lengths of the two sequences. It requires a local distance measure that can be used to compare individual feature vectors which should have small values when the vectors are similar and large values when they are different:

$$d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \geq 0 \tag{14.1}$$

The algorithm starts by building the distance matrix $C \in \mathbb{R}^{M \times N}$ representing all the pairwise distances between the feature vectors of the two sequences. The goal of the algorithm is to find the **alignment or warping** path which is a correspondence between the elements of the two sequences with the boundary constraint that the first and last elements of the two sequences are assigned to each other. Intuitively for matching sequences the alignment path will be roughly diagonal and will run through the low-cast areas of the cost matrix. More formally the alignment is a sequence of points $(p_i, p_j) \in [1 : M] \times [1 : N]$ for which the starting and ending points must be the first and last points of the aligned sequences, the points are time-ordered and each step size is constrained to either

move horizontally, vertically or diagonally. The cost function of the alignment path is the sum of all the pairwise distances associated with its points and the alignment path that has the minimal cost is called the **optimal alignment path** and is the output of the DTW.



(a) Good Alignment (b) Bad Alignment

Figure 14.1: *Similarity Matrix between energy contours and alignment path using Dynamic Time Warping*

Figure 14.1 shows two distance matrices that are calculated based on energy contours of different orchestra music movements. The left matrix is between two performances by different orchestras of the same piece. Even though the timing and duration of each performance is different they exhibit a similar overall energy envelope shown by the energy curves under the two axes. The optimal alignment path computed by DTW is shown imposed over the distance matrix. In contrast the matrix on the right shows the distance matrix between two unrelated orchestral movements where it is clear there is no alignment.

Cover song detection is performed by applying DTW between the query song and all the references and returning as a potential match the one with the minimum total cost for the optimal alignment path. Typically the alignment cost between covers of the same song will be significantly lower than the alignment cost between two random songs. DTW is a relatively costly operation and therefore this approach does not scale to a large number of songs. A common solution for large scale matching is to apply an audio fingerprinting type of approach with efficient matching to filter out a lot of irrelevant candidates and , once a sufficiently small number of candidate reference tracks have been selected, apply pair-wise DTW between the query and all of them.

	RE	SZA	TA
Mean # of covers in top 10	6.20	7.35	1.96
Mean Average Precision	0.66	0.75	0.20
Mean Rank of first correct cover	2.28	6.15	29.90

Table 14.1: 2009 MIREX Audio Cover Song Detection - Mixed Collection

	RE	SZA	TA
Mean # of covers in top 10	8.83	9.58	5.27
Mean Average Precision	0.91	0.96	0.56
Mean Rank of first correct cover	1.68	1.61	5.49

Table 14.2: 2009 MIREX Audio Cover Song Detection - Mazurkas

Table 14.1 shows the results of the audio cover song detection task of MIREX 2009 in the so called "mixed" collection which consists of 1000 pieces that contain 11 "cover songs" each represented by 11 different versions. As can be seen the performance is far from perfect but it is still impressive given the difficulty of the problem. An interesting observation is that the objective evaluation measures are not consistent. For example the RE algorithm performs slightly worse than the SZA in terms of mean average precision but has been mean rank for the first correctly identified cover. Table 14.2 shows the results of the MIREX 2009 audio cover song detection task for the Mazurkas collection which consists 11 different performances/versions of 49 Chopin Mazurkas. As can be seen from the results this is an easier dataset for which to find covers probably due to the smaller size and more uniformity in timbre. The RE algorithm is based on the calculation of different variants of chroma vectors utilizing multiple feature sets. In contrast to the more common approach of scoring the references in a ranked list and setting up a threshold for identifying covers it follows a classification approach in which a pair is either classified as reference/cover or as reference/non-cover. The SZA algorithm is based on harmonic pitch class profiles (HPCP) which are similar

to chroma vectors but computed over a sparse harmonic representation of the audio signal. The sequence of feature vectors of one song is transposed to the main tonality of the other song in consideration. A state space representation of embedding m and time delay z is used to represent the time series of HPCP with a recurrence quantification measure used for calculating cover song similarity.

Chapter 15

Other topics

In this chapter we cover several MIR topics that either are emerging or their associated literature is limited or both.

15.1 Optical Music Recognition

[16] [74] [**?**]

15.2 Digital Libraries and Metadata

[5], [26] [**?**], [70] [27]

15.3 Computational Musicology

15.3.1 Notated Music

[17] [61]

15.3.2 Computational Ethnomusicology

[35] [13] [?] [?] [?] [?]

15.3.3 MIR in live music performance

15.4 Further Reading

Part III

Systems and Applications

Chapter 16

Interaction

16.1 Interaction

[36]
[?]

Chapter 17

Evaluation

17.1 Bibliography

Chapter 18

User Studies

[?] [?] [?]

Chapter 19

Tools

Although frequently researchers implement their own audio feature extraction algorithms there are several software collections that are freely available that contain many of the methods described in this chapter. They have enabled researchers more interested in the data mining and machine learning aspects of music analysis to build systems more easily. They differ in the programming language/environment they are written in, the computational efficiency of the extraction process, their ability to deal with batch processing of large collections, their facilities for visualizing feature data, and their expressiveness/flexibility in describing complex algorithms.

Table 19.1 summarizes information about some of the most commonly used software resources as of the year 2010. The list is by no means exhaustive but does provide reasonable coverage of what is available. Several of the figures in this chapter were created using *Marsyas* and some using custom MATLAB code.

19.1 Datasets

[38]

Name	URL	Programming Language
Auditory Toolbox	tinyurl.com/3yomxwl	MATLAB
CLAM	clam-project.org/	C++
D.Ellis Code	tinyurl.com/6cvtdz	MATLAB
HTK	htk.eng.cam.ac.uk/	C++
jAudio	tinyurl.com/3ah8ox9	Java
Marsyas	marsyas.info	C++/Python
MA Toolbox	pampalk.at/ma/	MATLAB
MIR Toolbox	tinyurl.com/365oojm	MATLAB
Sphinx	cmusphinx.sourceforge.net/	C++
VAMP plugins	www.vamp-plugins.org/	C++

Table 19.1: Software for Audio Feature Extraction

19.2 Digital Music Libraries

VARIATIONS

Appendix A

Statistics and Probability

A brief overview of the necessary concepts.

Email Ashley to send me his PhD thesis

Appendix B

Project ideas

In this Appendix, I provide some information about projects related to music information retrieval based on the course that I have taught at the University of Victoria. First I provide some information about how to plan and organize a group project in MIR. A list of past projects done in the course is provided, followed by some additional suggestions. These are appropriate for class projects of small teams (2-3 students). I am really proud that some of these projects have resulted in ISMIR publications. Another good source of project ideas are the online proceedings of the ISMIR conference.

B.1 Project Organization and Deliverables

Music Information Retrieval is an interdisciplinary field so there is considerable variation in the types of projects one can do. The projects described in this appendix differ in the skills required, type of programming, availability of existing published work, and many other factors. The exact details are usually refined over the course of the term through better knowledge of MIR and interaction with the instructor. The expectations for each project are also adjusted depending on the number of students in the group.

Typically projects have the following components/stages:

1. Problem specification, data collection and ground truth annotation

2. Information extraction and analysis

3. Implementation and interaction

4. Evaluation

At least one of these components/stages should be non-trivial for each project. For example if the project tackles a completely new task then just doing a good problem specification and associated data collection might be sufficient, followed by some baseline analysis using existing tools. Another possibility would be to take an existing task for which there is data and code and build a really nice and intuitive graphical user interface in which case the non-trivial part would be the implementation and interaction stage. A thorough experimental comparison of existing techniques for a particular task would also be a valid project. What should be avoided is just taking existing tools, data for a well known task and just getting some results. Excellent projects which would have a good chance of being accepted as ISMIR submissions typically have more than one of these stages being novel and non-trivial. The exact details are worked out throughout the term based on the project deliverables.

The following deliverables are required for each project. All the written reports should conform to the template used for ISMIR submissions and follow the structure of an academic paper. It is a good idea to read a few ISMIR papers in order to familiarize yourself with both the template and the sections that are typically used. Each successive report builds upon the previous one. There is also a final presentation. In the MIR course at the University of Victoria, the project is worth 60% of the final grade.

1. **Design Specification** (15%)

   The design specification describes the proposed project, provides a timeline with specific objectives, and outlines the role of each team member. In addition it describes the tools, data sets, and associate literature that the group is going to use for the project. This report should be 2-4 pages using the ISMIR template. A proper bibliography of 15-20 references should also be provided.

2. **Progress Report** (15%)

   The progress report extends the design specification with specific information about what has been accomplished so far. Specific objectives achieved

are mentioned and deviations from the original plan are discussed. The time line is also revised to reflect the deeper understanding of the project. Additional literature can also be provided at this stage. This report should be 1-2 pages using the ISMIR template in addition to the text of the design specification.

3. **Final Report** (20%)

   The final report describes the system developed, the evaluation methodology, and the role of each member. It also describes which of the objectives were achieved, what has been learned and provides ideas for future work. The target audience is a future student of the course who would like to work on a similar project. The final report should be 2-3 pages of text in addition to the text of the progress report. The final document which will contain parts of the design specification, progress report and final report should be 6-10 pages using the ISMIR template. A bibliography of 20-30 references to literature relevant to the project should also be included.

4. **Presentation** (10%)

   The final presentation is typically done with all of the class attending and is a 10 minute slide presentation summarizing what was done and what was learned. It can also include a short video or demo of the system developed. For groups or students who can not attend physically the final presentation, a 10-minute video or audio recording+slides can be used instead.

B.2 Previous Projects

The following projects have been completed by groups of students in past offering of the Music Information Retrieval course at the University of Victoria. There is some variety in terms of difficulty and novelty but that is intentional as students can pick a project that is a more realistic match with their time commitment to the course. The descriptions are taken directly with minimal editing from the abstracts of the final project reports written by the students.

- **Physical Interfaces for Music Retrieval**

  This project explores integration of tangible interfaces with a music information retrieval system. The framework, called Intellitrance, uses sensor

data via midi input data to query and search a library of beats/sound/loops. This application introduces a new level of of control and functionality to the modern DJ and expert musician with a variety of flexible capabilities. IntelliTrance is written in Marsyas, a rapid prototyping and experimentation system that offers audio analysis and synthesis with specific emphasis to music signals and music information retrieval.

- **World Music Classification**

  Classification of music intro a general genre is a difficult task even for most humans; it is even more difficult to automate by a computer. Typically music would have attached to it meta-data that would include its genre classification, but as music databases become more prolific, this information is essentially missing. Automatic classification becomes essential for the organizing of music databases. In this project, we classify the genre of international music from a specific area and train a classifier to predict the genre of an unknown song given that it is from the same region. For this project we chose Turkish music as our region of interest.

- **Audio Caricatures**

  We have set out to tackle three problems. The first is how to accurately detect pitch in acoustic songs in wav format. The second is how to translate this information into a high level form that can be read and used by music production programs. And finally we want to create a caricature of the original wav file by excluding certain instruments. The result of our attempt at these problems has been Kofrasi, a perl program that uses text files produced by the pitch detection program, which bridges the gap between analyzing the original wav files and creating the final caricature.

- **An Interactive Query-by-Humming System**

  The challenge to create an effective and robust front-end for a query-by-humming system is a familiar topic in the field of Music Information Retrieval. This paper discusses the application we have created as a means to explore this problem. It first looks at the issues involved in creating an effective user interface and examines our own solution, specifically in relation to how it reconciles ease of use with maximal user control.

- **SIREN: A Sound Information Retrieval ENgine**

Multimedia players and the Internet have significantly altered the way people listen to and store music libraries. Many people do not realize the value of thier personal collections because they are unable to organize them in a clear and intuitive manner that can be efficiently searched. Clearly, a better way of fetching audio files and creating rich playlists from personal music collections needs to be developed utilizing MIR techniques. We have developed SIREN, a multimedia player that has been optimized to perform this task.

- **MIXMASTER: Automatic Musical Playlist Generation**

  Mixmaster is a playlist creator. Start and end songs are specified, as well as a time length. Then, a playlist is created by finding songs that approach the same similarity as the specified end song, while not deviating greatly from the original start and that last as long as the user wanted.

- **Musical Instrument Classification in Mixture of Sounds**

  This project attempts to evaluate the possibility of classifying musical instruments from an amplitude panned stereo or multi-channel mix. First the signal from each individual instrument is isolated by the frequency domain upmix algorithm proposed by Avendano and Jot. Then various features are extracted from the isolated instrument sound. The last step is to classify these features and try to determine what instrument(s) are present in the mixture of sound. The results show that the feature extraction and classification produce decent performance on single instrument, and as a whole, the classification system also produces reasonable result considering the distortion and loss of information introduced by unmix.

- **Ostitch: MIR applied to musical instruments**

  The paper discusses the use of MIR in computer music instruments. This paper proposed and implements a performance time MIR based instrument (Ostitch) that produces ”audio mosaics” or ”audio collages”. Buffering, overlapping and stiching (audio concatenation) algorithms are discusse‘d - problems around these issues are evaluated in detail. Overlapping and mixing algorithms are proposed and implemented.

- **Extracting Themes from Symbolic Music**

  This project investigates extracting the dominant theme from a MIDI file. The problem was broken into two tasks: track extraction and theme extrac-

tion. Two algorithms were developed for both tasks and a combination of these algorithms were tested on various MIDI files.

- **Singer Gender Identification**

  Automatic detection of whether a recording contains a male or female voice is an interesting problem. Ideally a computer could be programmed to recognize whether a singer is male or female in both monophonic and polyphonic audio samples. Artificial neural networks are a common technique for classifying data. Given enough training data and sufficient time to train, a neural network should be able to classify the gender of a voice in a recording with fairly high accuracy. This report outlines our results in building an artificial neural network that can determine the gender of a voice in both monophonic and polyphonic recordings.

- **Music information retrieval techniques for computer-assisted music mashup production**

  A music mashup (also know as a bootleg, boot, blend, bastard pop, smashup, and cut-up) is a style of music where two or more songs are mixed together to create a new song. Traditionally, mashup artists use trial and error, record keeping of tempos and keys of their song collection, and meticulous audio editing to create mashups. For this research project I will investigate the use of music information retrieval techniques to assist this production process by automatically determining tempo and keys of songs as well determining the potential of two songs to mix together by using these features in addition to other features such as frequency distribution.

- **Automatic Classification and Equalization of Music**

  For this project we designed a program for a stereo system which automatically classifies the current music being played and changes the equalization to best match the genre of music.

- **Location- and Temporally-Aware Intelligent Playlists**

  As location-aware mobile devices (PMPs, media-centric cell phones, etc.) are becoming increasingly ubiquitous, developers are gaining location data based on GPS, cellular tower triangulation, and Wi-Fi access point triangulation for use in a myriad of applications. I propose to create a proof-of-concept location-aware intelligent playlist for mobile devices (e.g. an Apple iPod Touch) that will generate a playlist based on media metadata (such as

tag, BPM, track length, etc.) and profiling the user's preferences based on absolute location as well as route of travel.

Route-of-travel based information could be useful in determine a user's listening habits while in transit to common destinations ("What does the user like listening to on the way to UVic from home?"), but another major factor missing is the time of travel. Does the user like listening to something mellow on the way to UVic in the morning, but prefer something more energetic while heading to UVic in the afternoon? If the user is heading downtown in the afternoon to shop, do they listen to the same music as when heading downtown at 10pm to hit a club? By profiling the user by absolute location, path, destination, and time, more interesting and appropriate playlists may be created using a history of the user's preference than a shotgun approach of using one playlist for the entire day without context.

- **Guitar Helper**

  A system is developed that detects which notes are being played on a guitar and creates a standard guitar tab. The system should work for both single notes and multiple notes/chords. An extension to the system would be to have the system perform accompaniment in the form of beats and/or chords. This will require beat detection and key detection.

- **RandomTracks : Music for sport**

  Random tracks is a GUI that allows the user to select a genre of music, or several genres of music to be played amongst each other. These genres can be played for a set period of time (for example perhaps 45 minutes of ambient music, or 3 minutes of punk rock). The specific problem this program will be built for is compiling a playlist for boxing sparring rounds. In gyms, music plays songs of various genres (mostly metal/rock) with the low points of the song sometimes coinciding with the more intense parts of training (the end of a 3 minute round). This program - randomtracks - can choose to play the most "intense" 3 minutes from a song that fits the genre specified by the user. As an example, the user may want to specify: A: 3 minutes, Rock. Increasing intensity. B: 1 minute, Ambient. Random start position. Alternate A and B. (Could also randomize A and B).

- **Robot Drummer Clap Tracking**

  Build a clap sensor - this will offload the signal processing from the Arduino to another chip and eliminate the need to move around the robot's

stepper motor wiring (the stepper is connected to the analog input pins). Measure the software delays and physical delays between the stimulus and the drummer's strike at various velocities, to get a table of delays that can be used to predict the next beat. Modify the robot's firmware to accept the clap sensor's digital output and predictively play along with a simple, regular beat pattern. Program the robot to play along with more sophisticated patterns (e.g. varations in timing and velocity).

- **Instrument mimicking synthesizer**

  We suggest a model which can approximately represent the characteristics of a musical instrument in one kilobit of data. We then describe how to extract this representation from a single recorded note of an instrument. Using these, we develop an instrument mimicking polyphonic midi synthesizer which produces recognizable output for a wide range of instruments.

- **Chipifier: Converting Music to Chiptune**

  This document will discuss the process required to convert a music file into a chiptune version. An overview of the required steps and algorithms will be covered. Initially, only a core set of the functionality will be implemented. Furthermore, the problem will be simplified by only accepting single instrument sounds. This will allow the core functionality to be developed with a simple goal. Additional features will then be added once the core is complete.

- **Low-power microcotrontroller real-time pitch detection for the guitar**

  This project has demonstrated that fundamental pitch (F0) approximation can be performed on a raw signal from an electric guitar using low-cost, low-power microcontrollers in real-time. To achieve this goal, a number of disciplines had to be understood; hardware design and construction, digital signal processing, hardware amplification/filters, low-level hardware programming, and Bayesian Classifiers. The PIC18F4620 is an 8-bit processor running at 20MHz that has onboard 8-bit hardware multipliers. To reduce processing, the sampling rate of the analog signal was limited to 2 KHz, meaning the highest note from the guitar for the purpose of this investigation has a F0 of 880 Hz (musical note A5). This was determined by giving a small margin of error to the Nyquist Theorem, which states the sampling rate can be at maximum one half of the highest frequency. The

passive low-pass filter used to remove aliased frequencies has a slow drop-off rate, so to avoid the possibility of aliasing, the maximum frequency was reduced from the theorems nominal value of 1 KHz. The lower sample rate drastically reduced the number of sample bins in the FFT calculation required for useable resolution in the frequency domain. Once the most dominate F0 signal had been disseminated using Bayesian classifiers from the signals FFT results, resynthesis of the frequency was done through two programmable sound generators (PSG).

B.3 Suggested Projects

The following are some representative ideas for projects for this class. They are listed in no particular order.

- **Feature Extraction on Multi-Core Processors**

  Feature extraction forms the basis of many MIR algorithms. It is a very computationally intensive process. However it has low memory requirements and is very straightforward to parallelize. The goal of this project is to explore how modern multi-core processor can be utilized to provide more efficient feature extraction. Experiments to find out what is the best granularity will be conducted using the Marsyas software framework.

- **Genre classification on MIDI data**

  Genre classification has mostly been explored in the audio domain for some time. More recently algorithms for genre classification based on statistics/features over symbolic data such as MIDI files have appeared in the literature. The goal of this project would be to recreate some of the existing algorithms and investigate alternatives.

- **A framework for evaluating similarity retrieval for music**

  A variety of similarity-based retrieval algorithms have been proposed for music in audio-format. The only way to reliably evaluate content-based similarity retrieval is to conduct user studies. The goal of this project is to build a framework (possibly web-based) that would allow different algorithms for audio similarity to be used and evaluated by users. The main challenge would be to design the framework to be flexible in the way the algorithms are evaluated, the similarity measure, the presentation mode etc.

- **Sensor-based MIR**

  One of the less explored areas in MIR is the interface of MIR systems to the user. As more and more music is available in portable digital music players of various forms and sizes we should envision how MIR can be used on these devices. This project is going to explore how sensor technology such as piezos, knobs, sliders can be used for browsing music collections, specifying music queries (for example tapping a query or playing a melody), and for annotation such onset detection and beat locations.

- **Key finding in polyphonic audio**

  There has been some existing work on key finding on symbolic scores. In addition, pitch-based representations such as Chroma vectors or Pitch Histograms have been shown to be effective for alignment, structural analysis and classification. This project will explore the use of pitch-based representations in order to identify the key in polyphonic audio recordings.

- **Query-by-humming front-end**

  The first stage in a QBH system is to convert a recording of a human singing, humming or whistling into either a pitch contour or note sequence that can then be used to search a database of musical pieces for a match. A large variety of pitch detection algorithms have been proposed in literature. This project will explore different pitch detection algorithms as well as note segmentation strategies

- **Query-by-humming back-end**

  Once either a pitch contour or a series of notes have been extracted they can be converted to some representation that can then be used to search a database of melodies for approximate matches. In this project some of the major approaches that have been proposed for representing melodies and searching melodic databases will be implemented.

- **ThemeFinder**

  In order to search for melodic fragments in polyphonic music it is necessary to extract the most important ”themes” of a polyphonic recording. This can be done by incorporating knowledge from voice leading, MIDI instrument labels, amount of repetition, melodic shape and many other factors. The goal of this project is to implement a theme finder using both techniques described in the literature as well as exploring alternatives.

- **Structural analysis based on similarity matrix**

  The similarity matrix is a visual representation that shows the internal structure of a piece of music (chorus-verse, measures, beats). By analyzing this representation it is possible to reconstruct the structural form of a piece of music such as AABA.

- **Drum pattern similarity retrieval**

  Drums are part of a large number of musical pieces. There are many software packages that provide a wide variety of drum loops/pattern that can be used to create music. Typically these large drum loop collections can only by browsed/searched based on filename. The aim of this project is to explore how the actual sound/structural similarity between drum patterns can be exploited for finding drum loops that are "similar". Accurate drum pattern classification/similarity can potentially lead to significant advances in audio MIR as most of recorded music today is characterized by drums and their patterns.

- **Drum detection in polyphonic audio**

  Recently researchers have started looking at the problem of identifying individual drum sounds in polyphonic music recordings such as hihat, bass drum etc. In this project, students will implement some of these new algorithms and explore variations and alternative approach. A significant part of the project will consists of building tools for obtaining ground truth annotations as well as evaluating the developed algorithms.

- **Content-based audio analysis using plugins**

  Many of the existing software music players such as WinAmp or itunes provide an API for writing plugins. Although typically geared toward spectrum visualization these plugins could pottentially be used as a front-end for feature extraction, classification and similarity retrieval. This project will expore this possibility.

- **Chord-detection in polyphonic audio**

  Even though polyphonic transcription of general audio is still far from being solved a variety of pitch-based representations such as chroma-vectors and pitch histograms have been proposed for audio. There is some limited research on using such representations potentially with some additional

knowledge (such as likely chord progression) to perform chord detection in polyphonic audio signals. The goal of this project is to explore possibilities in that space. Jazz standards or beatles tunes might be a good starting point for data.

- **Polyphonic alignment of audio and MIDI**

  A symbolic score even in a ”low” level format such as MIDI contains a wealth of useful information that is not directly available in the acoustic waveforms (beats/measures/chords etc). On the other hand most of the time we are interesting in hearing actual music rather than bad sounding MIDI files. In polyphonic audio alignment the idea is to compute features on both the audio and MIDI data and try to align the two sequences of features. This project will implement some of the existing approaches to this problem and explore alternatives and variations.

- **Music Caricatures**

  Even though we are still a long way from full polyphonic transcription music information retrieval are increasingly extracting more and more higher-level information about audio signals. The idea behind this project is to use this information to create musical ”caricatures” of the original audio using MIDI. The only constrain is that the resulting ”caricature” should somehow match possibly in a funny way the original music.

- **Comparison of algorithms for audio-segmentation**

  Audio segmentation referes to the process of detecting when there is a change of audio ”texture” such as the change from singing to instrumental background, the change from an orchestra to guitar solo, etc. A variety of algorithms have been proposed for audio segmentation. The goal of this project is to implement the main approaches and explore alternatives and variants.

- **Music Information Retrieval using MPEG-7 low level descriptors**

  The MPEG-7 standard was recently proposed for standarizing some of the ways multimedia content is described. Part of it describes some audio descriptors that can be used to characterize audio signals. There has been little evaluation of those descriptors compared to more other feature front-ends proposed in the literature. The goal of this project is to implement

the MPEG-7 audio descriptors and compare them with other features in a variety of tasks such as similarity retrieval, classification and segmentation.

- **Instrumentation-based genre/style classification**

  The type of instruments used in a song can be a quite reliable indicator of a particular musical genre. For example the significant presense of saxophone probably implies a jazz tune. Even though these rules always have exceptions they still will probably work for many cases. The goal of this project is to explore the use of decision trees for automatically finding and using such instrumentation-based rules. A significant part of the project will consist of collecting instrumentation annotation data.

- **Template-based detection of instrumentation**

  The goal of this project is to detect what (and maybe when) instruments are present in an audio recording. The goal is not source separation or transcription but rather just a presense/absence indicator for particular instruments. For example from minute 1 to minute 2 there is a saxophone, piano and drums playing after which a singer joins the ensemble would be the output of such a system. In order to identify specific instruments templates will be learned from a large database of examples and then adapted to the particular recording.

- **Singing-voice detection**

  Detecting the segments of a piece of music where there is singing is the first step in singer identification. This is a classic classification problem which is made difficult by the large variety of singers and instrumental backgrounds. The goal of this project is to explore various proposed algorithms and feature front-ends for this task. Specifically the use of phasevocoding techniques for enhancing the prominent singing voice is a promising area of exploration.

- **Singer Identification**

  The singer identity is major part of the way popular music is characterized and identified. Most listeners that hear a piece they haven't heard before can not identify the group until the singer starts singing. The goal of this project is to explore existing approaches to singer identification and explore variations and alternatives.

- **Male/Female singer detection**

  Automatic male/female voice classification has been explored in the context of the spoken voice. The goal of this project is to first explore male/female singer detection in monophonic recordings of singing and then expand this work to polyphonic recordings.

- **Direct manipulation music browsing**

  Although MIR for historical reasons has been mostly focused on retrieval a large part of music listening involves browsing and exploration. The goal of this project is to explore various creative ways of browsing large collections of music that are direct and provide constant audio feedback about the user actions.

- **Hyperbolic trees for music collection visualization**

  Hyperbolic trees are an impressive visualization technique for representing trees/graphs of documents/images. The goal of this project is to explore the potential of using this technique for visualizing large music collections. Of specific interest is the possibility adjustment of this technique to incorporate content-based music similarity.

- **Playlist summarization using similarity graphs**

  Similarity graphs are constructed by using content-based distances for edges and nodes that correspond to musical pieces. The goal of this project is to explore how this model could be used to generate summaries for music playlists i.e a short duration representation (3 seconds for each song in the playlist) that summarizes a playlist.

B.4 Projects that evolved into publications

Some of the best previous projects in the MIR course evolved into ISMIR publications (typically with some additional work done after the completion of the course). The corresponding publications are a good starting point and hopefully will inspire you to work hard on your projects with the eventual goal of an ISMIR publication.

Drum transcription for audio signals can be performed based on onset detection and subband analysis [104]. There are some interesting alternative ways

of specifying a music query beyond query-by-example and query-by-humming. One possibility is beat-boxing [**?**] and another is various interfaces for specifying rhythm [54]. Tabletop displays provides interesting possibilities for collaboration and intutive music browsing [46]. Stereo panning information is frequently ignored in MIR in which typically stereo signals are converted to mono. However stereo panning information is an important cue about the record production process and can be used for classifying recording production style [103]. Smart phones contain location and acceleration sensors that can be used ot infer the context of user activites and create personalized music based on the occassion [79]. Audio analysis can be used as an empirical analytical tool to study how DJs select and order tracks in electronic dance music [56].

Appendix C

Commercial activity 2011

Commercial activity in MIR related topics is in constant flux so it is hard to provide comprehensive information about it. At the same time it is interesting to observe which algorithms and techniques described in this book are actively used in real world applications. This appendix attempts to provide a snapshot of the current state of commercial activity in 202. The list of companies and systems described is by no means exhaustive but tries to be representative of existing activity.

Appendix D

Historic Time Line

In this Appendix I provide a list of historic milestones both academic and commercial in the emerging field of Music Information Retrieval. As is the case with any such attempt the resulting list is to some extent idiosyncratic. My hope is that it provides a picture of how the field evolved over time and helps readers situate the techniques and developments described in this book in time.

Bibliography

[1] E. Allamanche, J. Herre, O. Hellmuth, B. Froba, and T. Kastner. Content-based identification of audio material using MPEG-7 low level description. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2001.

[2] E. Allamanche, J. Herre, O. Hellmuth, B. Fröba, T. Kastner, and M. Cremer. Content-based identification of audio material using mpeg-7 low level description. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[3] J. J. Aucouturier and F. Pachet. Music similarity measures: What's the use ? In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[4] J. J. Aucouturier and F. Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):1–2, 2003.

[5] D. Bainbridge. The role of music ir in the new zealand digital music library project. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[6] J. Barthélemy. Figured bass and tonality recognition. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[7] M. A. Bartsch and G. H. Wakefield. To catch a chorus: using chroma-based represenations for audio thumbnailing. In *Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 150–18, 2001.

[8] J. P. Bello, G. Monti, and M. Sandler. Techniques for automatic music transcription. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[9] D. Bendor and M. Sandler. Time domain extraction of vibrato from monophonic instruments. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[10] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl. Aggregate features and AdaBoost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

[11] W. P. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. A. Bartsch, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[12] P. Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences*, 17:97–110, 1993.

[13] A. Bonardi. Ir for contemporary music: What the musicologist needs. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[14] A. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, 1990.

[15] M. Casey. Sound classification and similarity tools. In B. Manjunath, P. Salembier, and T. Sikora, editors, *Introduction to MPEG-7: Multimedia Content Description Language*, pages 309–323. J. Wiley, 2002.

[16] G. S. Choudhury, T. DiLauro, M. Droettboom, I. Fujinaga, B. Harrington, and K. MacMillan. Optical music recognition system within a large-scale digitization project. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[17] M. Clausen, R. Engelbrecht, D. Meyer, and J. Schmitz. Proms: A web-based tool for searching in polyphonic music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[18] D. Cliff and H. Freeburn. Exploration of point-distribution models for similarity-based classification and indexing of polyphonic music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[19] M. Crochemore, C. S. Iliopoulos, Y. J. Pinzon, and W. Rytter. Finding motifs with gaps. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[20] R. Dannenberg and N. Hu. Pattern discovery techniques for music audio. *Journal of New Music Research*, June 2003:153–164, 2003.

[21] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Math*, 41:909–996, 1988.

[22] S. Davis and P. Mermelstein. Experiments in syllable-based recognition of continuous speech. *IEEE Transcactions on Acoustics, Speech and Signal Processing*, 28:357–366, Aug. 1980.

[23] A. de Cheveigne and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of Acoustical Society of America*, 111(4):1917–1930, 2002.

[24] S. Dixon. Onset detection revisited. In *Int. Conf. on Digital Audio Effects (DAFx)*, 2006.

[25] J. S. Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[26] J. W. Dunn. Beyond variations: Creating a digital music library. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[27] J. W. Dunn, M. W. Davidson, and E. J. Isaacson. Indiana university digital music library project: An update. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[28] L. D. E. Ravelli, G. Richard. Audio signal representations for indexing in the transform domain. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(3):434–446, 2010.

[29] D. Ellis and G. H. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages IV–1429–IV–1432, 2007.

[30] M. T. F. Morchen, A. Ultsch and I. Lohken. Modeling timbre distance with temporal statistics from polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 8(1):81–90, 2006.

[31] J. Foote. Visualizing music and audio using self-similarity. In *ACM Multimedia*, pages 77–80, 1999.

[32] J. Foote. Arthur: Retrieving orchestral music by long-term structure. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[33] J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[34] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno. A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre similarity based music information retrieval. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):638–648, 2010.

[35] G. Geekie. Carnatic ragas as music information retrieval entities. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2002.

[36] A. Georgaki, S. Raptis, and S. Bakamidis. A music interface for visually impaired people in the wedelmusic environment. design and architecture. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[37] M. Good. Representing music using xml. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[38] H. . N. T. . O. R. Goto, Masataka ; Hashiguchi. Rwc music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2002.

[39] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *AES 25th Int. Conf. on Metadata for Audio*, pages 196–204, 2002.

[40] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on Audio, Speech and Language Processing*, 14(5):1832–1844, 2006.

[41] J. Grey. Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977.

[42] M. Gruhne, C. Dittmar, and D. Gaertner. Improving rhythmic similarity computation by beat histogram transformations. In *Int. Conf. on Music Information Retrieval*, 2009.

[43] S. Hainsworth. Beat tracking and musical metre analysis. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, pages 101–129. Springer, New York, 2006.

[44] T. Haitsma, Jaap ; Kalker. A highly robust audio fingerprinting system. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2002.

[45] P. Herrera-Boyer, X. Amatriain, E. Batlle, and X. Serra. Towards instrument segmentation for music content description:a critical review of instrument classification techniques. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[46] S. Hitchner, J. Murdoch, and G. Tzanetakis. Music browsing using a tabletop display. In *ISMIR*, pages 175–176, 2007.

[47] A. Holzapfel and Y. Stylianou. Rhythmic similarity of music based on dynamic periodicity warping. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2217–2220, 2008.

[48] A. Holzapfel and Y. Stylianou. A scale transform based method for rhythmic similarity of music. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 317–320, 2009.

[49] H. H. Hoos, K. Renz, and M. Görg. Guido/mir an experimental musical information retrieval system based on guido music notation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[50] Ö. Izmirli. Using a spectral flatness based feature for audio segmentation and retrieval. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[51] J. Jensen, M. Christensen, and S. Jensen. A tempo-insensitive representation of rhythmic patterns. In *European Signal Processing Conference (EUSIPCO)*, 2009.

[52] J. H. Jensen, D. Ellis, M. Christensen, and S. H. Jensen. Evaluation of distance measures between gaussian mixture models of mfccs. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2007.

[53] D. Jiang, L. Lu, H. Zhang, and J. Tao. Music type classification by spectral contrast feature. In *Int. Conf. on Multimedia and Expo (ICME)*, 2002.

[54] A. Kapur, R. I. McWalter, and G. Tzanetakis. New music interfaces for rhythm-based retrieval. In *ISMIR*, pages 130–136. Citeseer, 2005.

[55] M. Katz. *Capturing sound: how technology has changed music*. Univ of California Press, 2010.

[56] T. Kell and G. Tzanetakis. Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction. In *ISMIR*, pages 505–510, 2013.

[57] F. J. Kiernan. Score-based style recognition using artificial neural networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[58] Y. E. Kim, W. Chai, R. Garcia, and B. Vercoe. Analysis of a contour-based representation for melody. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[59] Y. E. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[60] T. Kitahara, Y. Tsuchihashi, and H. Katayose. Music genre classification and similarity calculation using bass-line features. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, 2008.

[61] A. Kornstädt. The jring system for computer-assisted musicological analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[62] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linnery. Classification of audio signals using statistical features on time and wavelet transform domains. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[63] C. Lee, J. L. Shih, K. Yu, and H. Lin. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Trans. on Multimedia*, 11(4):670–682, 2009.

[64] J. S. . R. A. Lee, Jin Ha ; Downie. Representing traditional korean music notation in xml. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2002.

[65] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio. *IEEE Trans. on Audio, Speech and Language Processing*, 16(2):291–301, 2008.

[66] T. Li and M. Ogihara. Detecting emotion in music. In *Int. Symposium on Music Information Retrieval (ISMIR)*, pages 239–240, 2003.

[67] T. Li and M. Ogihara. Toward intelligent music information retrieval. *IEEE Trans. on Multimedia*, 8(3):564–574, 2006.

[68] T. Li and G. Tzanetakis. Factors in automatic musical genre classification. In *Proc. Workshop on applications of signal processing to audio and acoustics WASPAA*, New Paltz, NY, 2003. IEEE.

[69] C. Liem and A. Hanjalic. Cover song retrieval: a comparative study of system component choices. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2009.

[70] K. Lin and T. Bell. Integrating paper and digital music information systems. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[71] A. Lindsay and Y. E. Kim. Adventures in standardization, or, how we learned to stop worrying and love mpeg-7. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[72] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[73] L. Lu, D. Lie, and H.-J. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Trans. on Speech and Audio Processing*, 14(1):5–18, 2006.

[74] K. MacMillan, M. Droettboom, and I. Fujinaga. Gamera: A structured document recognition application development environment. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[75] M. Mandel and D. Ellis. Song-level features and svms for music classification. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[76] M. F. McKinney and J. Breebaart. Features for audio and music classification. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[77] C. Meek and W. P. Birmingham. Thematic extractor. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[78] A. Meng, P. Ahrendt, and J. Larsen. Temporal feature integration for music genre classification. *IEEE Transactions on Audio, Speech and Language Processing*, 5(15):1654–1664, 2007.

[79] S. Miller, P. Reimer, S. R. Ness, and G. Tzanetakis. Geoshuffle: Location-aware, content-based music browsing using self-organizing tag clouds. In *ISMIR*, pages 237–242, 2010.

[80] V. Moorefield. *The producer as composer: Shaping the sounds of popular music*. MIT Press, 2010.

[81] G. T. N. Hu, R. B. Dannenberg. Polyphonic audio matching and alignment for music retrieval. In *Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[82] T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka. Music signal spotting retrieval by a humming query using start frame feature dependent continuous dynamic programming. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[83] T. Painer and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000.

[84] Y. Panagakis, C. Kotropoulos, and G. C. Arce. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):576–588, 2010.

[85] J. Paulus and A. Klapuri. measuring the similarity of rhythmic patterns. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[86] G. Peeters. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *IEEE Trans. on Audio, Speech, and Language Processing*, (to appear), 2010.

[87] G. Peeters, A. L. Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[88] D. Pye. Content-based methods for management of digital music. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2000.

[89] A. Rauber, E. Pampalk, and D. Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, 2003.

[90] P. Roland. Xml4mir: Extensible markup language for music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[91] B. D. S. Essid, G. Richard. Musical instrument recognition by pairwise classification. *IEEE Trans. on Audio, Speech and Language Processing*, 14(4):1401–1412, 2006.

[92] J. Serr and E. Gmez. Audio cover song identification based on tonal sequence alignment. In *IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, pages 61–64, 2008.

[93] A. Sheh and D. P. W. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[94] H. Sotlau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[95] P. Tschmuck. *Creativity and Innovation in the Music Industry*. Springer, 2012.

[96] E. Tsunoo. Audio genre classification using percussive pattern clustering combined with timbral features. In *Int. Conf. on Multimedia and Expo (ICME)*, 2009.

[97] E. Tsunoo, N. Ono, and S. Sagayama. Musical bass-line pattern clustering and its application to audio genre classification. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2009.

[98] G. Tzanetakis and P. Cook. Multi-feature audio segmentation for browsing and annotation. In *Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999.

[99] G. Tzanetakis and P. Cook. Audio information retrieval (air) tools. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2000.

[100] G. Tzanetakis and P. Cook. Sound analysis using mpeg compressed audio. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2000.

[101] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[102] G. Tzanetakis, R. Jones, and K. McNally. Stereo panning features for classifying recording production style. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2007.

[103] G. Tzanetakis, R. Jones, and K. McNally. Stereo panning features for classifying recording production style. In *ISMIR*, pages 441–444, 2007.

[104] G. Tzanetakis, A. Kapur, and R. I. Mcwalter. Subband-based drum transcription for audio signals. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, pages 1–4. IEEE, 2005.

[105] G. Tzanetakis, L. Martins, K. McNally, and R. Jones. Stereo panning information for music information retrieval tasks. *Journal of the Audio Engineering Society*, 58(5):409–417, 2010.

[106] K. West and S. Cox. Finding an optimal segmentation for audio genre classification. In *Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.