TU WIEN

Technische Universität Wien
Vienna University of Technology

DIPLOMARBEIT

Evaluation of New Audio Features and Their Utilization in Novel Music Retrieval Applications

Ausgeführt am Institut für
Softwaretechnik und Interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von
ao. Univ.Prof. Dr. Andreas Rauber
Favoritenstraße 9-11/188
A-1040 Wien, AUSTRIA

durch

Thomas Lidy
Johann-Strauß-Gasse 24/20
A-1040 Wien

Dezember 2006

Acknowledgements

I wish to thank Andreas Rauber for enabling me to work on Music
Information Retrieval (MIR) and for his ongoing support and motivation.
I thank Elias Pampalk for being a continuous source of inspiration.
I particularly thank the people in MIR research for
integrating me cordially into the community.

I thank my parents for giving me the opportunity to study at University.
I thank all my friends for their patience, with special thanks to Emanuel.
Thanks goes to all the people who told me that it is *not* important
to finish one's studies in the minimum time necessary.

Zusammenfassung

Die wachsende Popularität und Größe von Musikarchiven – sowohl im privaten als auch im professionellen Bereich – erfordert neue Methoden für das Organisieren und Suchen von Musik sowie den Zugriff auf diese Musikkollektionen. Music Information Retrieval ist ein junges Forschungsgebiet, das sich mit der Entwicklung von automatischen Methoden zur Berechnung von Ähnlichkeit in Musik beschäftigt, um das Organisieren von großen Musikarchiven auf Basis von akustischer Ähnlichkeit zu ermöglichen. Für Musikähnlichkeit spielt eine Vielzahl an Aspekten eine Rolle: z.B. Tempo, Rhythmus, Melodie, Instrumentierung und potenziell auch die Struktur (Refrain und Vers), der Text und sogar die verwendete Sprache. Um Musik semantisch erfassen zu können, ohne jeden einzelnen Song manuell beschriften zu müssen, wird viel Forschung zur automatischen Extraktion solcher musikalischen Aspekte betrieben.

Diese Algorithmen zur sogenannten Feature (Merkmals-) Extraktion bilden das Herzstück einer Reihe von weiteren Aufgaben. Unter Verwendung von Klassifikationsalgorithmen können damit ganze Musikarchive automatisch in Kategorien organisiert werden. Allerdings stellt oft die Einteilung dieser Kategorien selbst ein Problem dar, sodass andere Methoden gefunden wurden, die Musiksammlungen rein aufgrund von Musikähnlichkeiten in Cluster gruppieren. Dabei wird Musik, die sehr ähnlich klingt, zusammen gruppiert und gleichzeitig von Musik mit anderen Charakteristika distanziert. Um das Resultat intuitiv darstellen zu können, wurde eine Reihe von Visualisierungen für die Darstellung von Musikarchiven entwickelt.

Diese Diplomarbeit stellt zwei neue Algorithmen für die automatische Merkmalsextraktion aus Musik vor und beschreibt eine Reihe von Verbesserungen an einem weiteren, bereits existierenden Verfahren. Weiters beinhaltet die Arbeit eine Studie zur Bedeutung der Psycho-Akustik in der Berechnung von Musikmerkmalen. Alle neuen Verfahren werden anhand von Referenz-Musikkollektionen sowie in internationalen Performancevergleichen (auf Basis von Genre-Klassifizierung, Interpret-Erkennung und Ähnlichkeitssuche) evaluiert. Darüber hinaus wird eine neuartige Software vorgestellt, die Musiksammlungen auf Musiklandkarten darstellt und das Finden ähnlicher Musik sowie die direkte Interaktion mit der Sammlung ermöglicht, und zwar sowohl auf PCs als auch auf mobilen Geräten. Zur Veranschaulichung wurden Mozarts gesamte Werke unter Verwendung der neuen Methoden zur Merkmalsberechnung auf einer Musiklandkarte organisiert und die *Map of Mozart* erstellt.

Abstract

With increased popularity and size of music archives – in both the private and professional domains – new ways for organizing, searching and accessing these collections are needed. Music Information Retrieval is a relatively young research domain which addresses the development of automated methods for computation of similarity within music, in order to enable similarity-based organization of large music archives.

In music similarity many different aspects play a role, e.g. tempo, rhythm, melody, instrumentation, but potentially also the structure (chorus/verse), the lyrics and even the language of a song. Much research is done on the automatic extraction of those aspects in order to describe music semantically, without the need of manual annotation.

Those feature extraction algorithms form the basis for a range of further tasks. Automatic organization of entire music archives into categories can be accomplished by the use of classification algorithms. However, often the definition of categories is a problem itself and thus methods have been created to cluster music collections solely by sound similarity. Clustering means that music which is very similar is grouped together and separated from music containing different characteristics. Visualizations have been devised to provide intuitive views of clustered music collections.

This work contributes two new algorithms for automatic extraction of features from music and presents a number of improvements on an existing descriptor. It contains a study on the importance of considering psychoacoustics in feature computation. The new approaches are evaluated on a number of reference music collections as well as in international benchmarking events on music genre classification, artist recognition and similarity retrieval.

Moreover, a set of novel applications for clustering music libraries on *Music Maps* is presented, allowing interaction with and retrieval of music both on personal computers and mobile devices. For demonstration of practicability Mozart's complete works have been organized on a Music Map, the *Map of Mozart*, which has been created utilizing the previously evaluated audio descriptors.

Contents

Chapter 1

Introduction

1.1 Motivation

Music has become one of the predominant goods in our world, not only, but with increasing importance, in the Internet. Digital music databases are continuously gaining popularity both in terms of professional repositories and personal audio collections. Broadcast stations, movie industry, national archives, etc. are among the professionals concerned with large audio databases. Ongoing advances in network bandwidth and popularity of Internet services anticipate further growth of the number of private people having large music collections in digital form.

However, the organization of large music collections is a very time-intensive and tedious task, especially when the traditional solution of manually annotating semantic data to the audio is chosen. Also, one cannot rely on meta-data services which deliver data such as artist name, title and album, because these meta-data may be incorrect or incomplete. Moreover, traditional search based on file name, song title, or artist does not meet the advanced requirements of people working with large music archives, because it either presumes exact knowledge of the meta-data fields or involves browsing of long lists in the archive. For many audio titles and archives apart from popular music such meta-data is not even available yet. Consequently, the possibility to search and organize music according to similarity inherent in the music itself is required.

Fortunately, the research domain of Music Information Retrieval (MIR) has made substantial progress in recent years to find solutions to these challenges. Approaches from Music Information Retrieval accomplish content-based analysis of music in order to automatically extract semantic descriptors. These descriptors are intended to capture significant aspects of music such as pitch, timbre, instrumentation, structure, tempo, beat, etc. However, it is an unsolved problem of what exactly to capture for efficient description of music. The descriptors, or features, extracted from music are fundamental to tasks like searching music (i.e. retrieval of similar music to a given piece), music identification, classification of music into categories (i.e. automatic meta-data labeling) and organization of music collections by similarity. The choice of features to extract is a matter of the specific task, but is also a matter of ongoing research. For instance the utilization of psycho-acoustics in feature extraction has not yet been investigated exhaustively.

In the music feature extraction domain there are two main directions: extraction from symbolic notations (e.g. MIDI files) and extraction from audio waveform signals (such was CDs, WAV or MP3 files). The challenge of the signal-based approaches is that only a mixed signal is available, containing all kinds of sources (different types of instruments, percussion, singing voice contained in a mixed signal). The challenge of notation-based approaches is that they do not have information about the actual sound of the music. This thesis has its focus entirely on systems that are based on extraction from audio signals rather than symbolic notations.

The extracted features form the basis for a range of applications. One of them is the automatic classification of music archives into a set of categories. Musical genre is probably the most popular categorization of music, promoted by the music industry, used in shops for the arrangement of CDs and also by home users to organize their music collections. Consequently, there is substantial need for automatic classification of music into genres. However, there is the open question of the definition of a genre. The actual genre categorization depends on the audio collection under consideration and/or the user's taste and experience. Nevertheless, with the use of classification techniques from the machine learning domain combined with recent approaches for music feature extraction considerable achievements have been made on music classification. Using these techniques, entire music archives can be classified and organized automatically.

Yet, musical genres are often defined in a fuzzy way and many genres are overlapping. Therefore, other approaches for the organization of music archives rely on clustering or topology-preserving mapping techniques, such as, for example, the Self-Organizing Map (SOM), which do not consider categories but organize music solely by perceived sound similarity. In clustering, the features extracted from music are analyzed and similar pieces of music are grouped together, forming clusters. Each type of music with a distinct style is represented by a separate cluster. Music which is not represented by a clearly defined style may be either assigned to or located between two or more clusters of more representative music, depending on the clustering approach used. The result is an overall organization of a music archive. In order to depict the clusters a number of visualization techniques have been devised which provide intuitive views of the music collection. Based on one of these visualizations the metaphor of a geographic map has been created for the representation of a music archive in order to create virtual landscapes of music collections, so-called *Music Maps*.

On top of these maps various applications have been developed, which offer completely novel ways of interaction with music archives: Music can be accessed and played directly based on similarity, while the overview of the whole music collection is preserved. Playlists can be created depending on a particular situation or mood, by simply selecting areas on the *Music Map*. This novel metaphor for retrieval and browsing of music is particularly useful on mobile devices and therefore efforts are made for the implementation of the new interaction models on handheld devices.

To summarize, approaches developed within the Music Information Retrieval research domain relieve us from the burden of manual annotation and labeling of music collections and of organizing them into categories. They enable us to find music based on the perceived sound similarity rather than unreliable or inexistent meta-data, they allow the identification of pieces of music, and they pave the way for completely new models of organization of and interaction with music collections.

1.2 Outline

This thesis is organized as follows:

Chapter 2 reviews related publications within areas relevant to the work in this thesis, such as feature extraction approaches from audio, music classification, clustering approaches, visualizations and interfaces for Music Information Retrieval (MIR).

In Chapter 3 several classes of standard audio features are explained, including MPEG-7 and MARSYAS features, as well as three novel feature sets that are evaluated and utilized in later chapters of this thesis.

Chapter 4 introduces the reference audio collections used in the various benchmarking campaigns and experiments described in this thesis.

Chapter 5 reviews the history of international benchmarking in MIR research and reports about both scientific evaluation campaigns as well as individual evaluations of the audio feature sets developed as part of this thesis work on both classification and similarity retrieval tasks.

Chapter 6 explains how music is clustered on Self-Organizing Maps, describes a large number of visualization methods for Music Maps and presents novel applications for interaction with music. The *Map of Mozart* is presented as a demonstration of the practicability of the audio feature extraction and map organization approaches.

Chapter 7 provides a summarization and draws conclusions.

1.3 Contributions

These are the contributions of this thesis:

- a review of common audio features for Music Information Retrieval tasks including MPEG-7 standard descriptors as well as state-of-the-art non-standardized feature sets

- an improved version of the Rhythm Pattern audio feature set, derived from an evaluation of the impact of utilizing psycho-acoustics in audio feature computation

- two new feature sets for audio content description: Statistical Spectrum Descriptors and Rhythm Histograms, which show comparable performance or even outperform Rhythm Patterns, yet at much lower dimensionalities of the resulting feature space

- benchmark evaluation of the feature sets on three standard music databases and evaluation of combinations of feature sets for music classification tasks

- participation in international evaluation campaigns starting from the ISMIR Audio Description Contest in 2004 up to the most recent MIREX evaluation campaign 2006 with results stating that the feature sets developed in the course of this thesis are amongst the best-performing state-of-the-art methods
- contributions to novel applications for visualization of and interaction with music collections on desktop computers (PlaySOM) and mobile devices (PocketSOMPlayer)
- utilization of the novel feature sets and applications for clustering of music archives using Self-Organizing Maps, with the particular example of clustering the complete works of W. A. Mozart and creation of the interactive “Map of Mozart”

Several parts of the research done for this thesis have been already reviewed and published at international scientific conferences. The new audio feature sets as well as related evaluation experiments were published and presented at the International Conference on Music Information Retrieval (ISMIR) 2005 [LR05]. A resynthesis approach of the Rhythm Pattern feature set has been demonstrated at the International Computer Music Conference (ICMC) 2005 [LPR05]. The submission to the ISMIR 2004 Audio Description Contest won a prize in the category of Rhythm Classification. The feature sets devised have been benchmarked in several international evaluation campaigns (MIREX). The Rhythm Pattern feature set has also been applied successfully to instrument classification [BKLR06]. The application of the feature sets to the clustering of music collections within the PlaySOM and PocketSOMPlayer programs has been presented at the Musicnetwork Workshop 2005 [NLR05].

Amongst the applications developed on top of the principles described in this thesis was the Map of Mozart, which was made public in spring 2006[1]. National and international online press (ORF Futurezone, Spiegel online [MoM06d], La Capital, et al.) as well as print media (Kurier [MoM06a], Der Standard [MoM06c], Financial Times Germany [MoM06e], GEO [MoM06b]) reported about the clustering of Mozart’s complete works on the Map of Mozart[2]. The Map of Mozart was also presented at ISMIR 2006 [MLR06].

[1] http://www.ifs.tuwien.ac.at/mir/mozart/

[2] http://www.ifs.tuwien.ac.at/mir/press.html

Chapter 2

Related Work

2.1 Introduction

The domain of content-based music retrieval experienced a major boost in the late 1990's when mature techniques for the automated description of the content of music became available. From that time on a growing number of researchers has been working on different methods for description, retrieval and organization of music based on its content. Many different approaches exist for computation of features from the musical content. Descriptors can be computed from music stored either in audio waveform signals (e.g. WAV or MP3 format) or in symbolic notations, which do not actually contain any sound (such as MIDI files). Music stored in symbolic notations is not part of this thesis and thus will not be considered in the following review of music descriptors in Section 2.2, which will consequently concentrate on feature extraction approaches from audio signals.

Orio explains and reviews different aspects of music and music processing, in both the audio and symbolic domains [Ori06]. He furthermore discusses the role of the users, describes several systems for music retrieval, browsing and visualization and gives an introduction to scientific Music Information Retrieval evaluation campaigns.

Stephen Downie provides a review of all aspects of Music Information Retrieval (MIR), covering as well all the individual classes of music descriptors [Dow03a]. He also discusses challenges in Music Information Retrieval and reviews different MIR systems.

Apart from a wealth of audio descriptors, a large range of different similarity measures have been published, which are partly mentioned either directly together with the audio descriptors in Section 2.2 (if they are systematically connected) or in Section 2.3 which reviews machine learning and classification approaches in MIR.

Section 2.4 describes the efforts for scientific evaluation in MIR and the creation of benchmark audio collections. Section 2.5 introduces a number of clustering approaches that emerge into different kinds of attractive visualizations, on top of which novel interfaces for access to music collections are created. A summary is given in Section 2.6.

2.2 Audio Feature Extraction

In the domain of feature extraction from audio a lot of different approaches have been developed. The wealth of devised audio descriptors include musical aspects such as loudness, tempo, beat, rhythm, timbre, pitch, harmonics, melody, etc. This list already included several higher-level descriptors which themselves are based on low-level features. The low-level features are directly based on the temporal or spectral representation of the audio signal, and some of them will be explained in detail in Chapter 3.

The following is a review of important works on the development of low-level and higher-level audio descriptors:

The first beat detection systems were already published in the 1970s and 1980s [Ste77, LH78, LHL82, CMRR82, DH89]. In 1990 Paul Allen and Roger Dannenberg presented a new approach for beat tracking working in real-time [AD90]. Contrary to previous beat detection algorithms their approach was adaptive, i.e. it predicts beats considering multiple interpretations of the performance. In 1995 Goto and Muraoka [GM95] propose another real-time beat tracking system for audio signals.

In 1996 Wold et al. present an audio analysis, search, and classification engine called Muscle Fish[1]. The system is intended for retrieval of sounds rather than music and uses features such as loudness, pitch, brightness and bandwidth [WBKW96].

Scheirer introduces a vocoder-based approach for tempo and beat analysis of musical signals [Sch98]. He presents a method for using a small number of bandpass filters and banks of parallel comb filters to analyze the

[1] http://www.musclefish.com

tempo of and extract the beat from musical signals of arbitrary polyphonic complexity. Scheirer did also a comparison of his vocoder model with the perceptually-based pitch model of Meddis and Hewitt [MH91], and discovered that the problems of pitch and pulse detection are related and that a pitch tracker can also be used for extracting the tempo of acoustic signals, yet at a larger time scale [Sch97].

More recent work on beat tracking includes the work of Dixon [Dix99]. A review of automatic rhythm description systems has been published in [GD05].

Mel-Frequency Cepstral Coefficients (MFCC) are a perceptual motivated set of features developed in context of speech recognition. An investigation about their adoption in the MIR domain was presented by Logan [Log00]. Liu and Huang [LH00] introduce a segmentation approach for audio based on MFCC features. They use Gaussian Mixture Models (GMM) to model feature distribution of an audio segment and propose the Kullback Leibler divergence [CT91] as a metric for distance measuring between two models, which was new to MIR research. Logan and Salomon [LS01] perform content-based audio retrieval based on K-Means clustering of MFCC features and apply yet another distance measure: the Earth Mover's Distance [RTG98].

Rauber, Pampalk and Merkl propose "Rhythm Patterns" [RPM03, PRM02], modeling modulation amplitudes on critical frequency bands, for organization and visualization of music archives. The approach is based on a previous development by Rauber and Frühwirth [RF01]. The new feature set consideres a set of psycho-acoustic models [RPM02]. A resynthesis algorithm of the Rhythm Patterns feature set allowing to analyze its characteristics has been shown later [LPR05].

Aucouturier and Pachet introduce a timbral similarity measure based on Gaussian Mixture Models of MFCCs [AP02], but also question the use of such measures in very large databases and propose a measure of "interestingness".

Pampalk et al. [PDW03] conduct a comparison of several content-based audio descriptors and similarity measures on both small and large audio databases, including those of Logan and Salomon [LS01] and Aucouturier and Pachet [AP02] as well as Rhythm Patterns (Fluctuation Patterns). They report that in the large scale evaluation simple Spectrum Histograms outperform all other descriptors. The various approaches have been extended, optimized, compared and combined in Pampalk's PhD thesis [Pam06], in which also a number of applications are presented.

Li et al. [LOL03] propose Daubechies Wavelet Coefficient Histograms as a feature set suitable for music genre classification. The feature set characterizes amplitude variations in the audio signal.

The MPEG-7 standard also defines a set of seventeen low-level audio descriptors [Mar04, MPE02]. Allamanche et al. [AHH$^+$01] use these features for audio fingerprinting, i.e. the robust identification of audio material. A classification approach with MPEG-7 features is done in [CW03]. Xiong et al. did a comparison of MFCC and MPEG-7 features on sports audio classification [XRDH03].

Liu and Tsai [LT01] propose an idea to derive audio features directly from the coefficients of the output of the polyphase filters of MP3-encoded music and an approach on content-based similarity based on that idea.

George Tzanetakis reviews in his PhD thesis [Tza02] a number of systems for the analysis and manipulation of audio data as well as content-based retrieval. He developed a new system for audio feature extraction, analysis and classification, called MARSYAS, which is designed for rapid prototyping of MIR research. He also contributed a number of new algorithms for audio description: a general multifeature audio texture segmentation methodology, feature extraction from MP3 compressed data (similar to the idea of Liu and Tsai), beat detection based on the discrete Wavelet transform and musical genre classification combining timbral, rhythmic and harmonic features. Furthermore he presents novel 2D and 3D graphical user interfaces for browsing and interacting with audio signals and collections (c.f. Section 2.5).

In a previous work Tzanetakis et al. [TEC02b] proposed Pitch Histograms as a way to represent the pitch content of music signals. Pitch is a feature yet mostly used in symbolic music description. This new approach was applicable to both symbolic and audio data. For the audio case a multiple-pitch detection algorithm for polyphonic signals by Tolonen and Karjalainen [TK00] is used to calculate the Pitch Histograms.

High-level audio feature sets include the extraction of key [Lem95, MMB$^+$05], tonality [Gom06] and melody [GKM03]. Purwins [Pur05] did an extensive study of pitch classes based on the circularity of relative pitch and key.

Another detailed review about audio descriptors is provided in Pohle's thesis [Poh05].

2.3 Music Classification

Many feature extraction algorithms are employed in the context of classification tasks, such as music/speech discrimination, classification of animal or environmental sounds [Mit05], music genre classification, or dance rhythm classification.

In 1997 Scheirer and Slaney present a speech/music discriminator [SS97] that is based on temporal and spectral low-level features such as percentage of low-energy frames, Zero-Crossing Rate, Spectral Rolloff Point, Spectral Centroid, and Spectral Flux. The system's performance is evaluated by classification using k-Nearest Neighbor (k-NN), Gaussian Mixture Models and a k-d tree. A Gaussian Mixture Model (GMM) models each class of data as the union of several Gaussian clusters in the feature space. This clustering can be iteratively derived with the Expectation Maximization (EM) algorithm [Moo96]. Classification using the GMM uses a likelihood estimate for each model, which measures how well the new data point is modeled. A point in feature space is assigned to whichever class is the best model of that point [SS97].

An early work on musical style recognition by Dannenberg et al. [DTW97] investigates various machine learning techniques applied for building style classifiers. The authors use low-level features from MIDI data and compare a Bayesian classifier, a linear classifier and neural networks for discriminating between the styles "lyrical", "frantic", "syncopated" and "pointilistic".

In a seminal work about "Content-Based Retrieval of Music and Audio" [Foo97] Foote uses MFCC features and proposes a tree-based supervised vector quantization approach. As distance measures the Euclidean and the cosine distance are compared, where the latter performs better. Foote also proposes the idea of directly using MPEG encoded data for feature extraction for the first time.

Liu and Wan [LW01] conduct a study in which four classifiers, namely nearest neighbor, modified k-NN, GMM and probabilistic neural networks were compared. Based on a small set of features, selected by feature selection, the task was to classify different types of sounds, speech, and music.

Tzanetakis et al. [TEC01] perform a hierarchical genre classification approach using and proposing a set of features representing texture and instrumentation in music. Li et al. [LOL03] conduct a comparative study on content-based music genre classification using several classifiers, including Support Vector Machines. Basili et al. [BSS04] present another study on different machine learning algorithms (and varying dataset partitioning) and their performance in music genre classification. Livshin and Rodet [LR03] present a cross database evaluation on musical instrument classification.

Dixon et al. [DPW03] employ Periodicity Patterns for classification of Latin American and Ballroom dance music. Gouyon and Dixon [GD04] propose also a tempo-based approach for dance music classification.

A survey on automatic genre classification of music content is available in [SZM06]. Details about machine learning and pattern classification algorithms are provided in [DHS00].

2.4 Benchmarking in MIR Research

Benchmarking is an important topic when comparing the many different audio descriptors and approaches for classification or music similarity computation.

A fundamental work for benchmarking was the large-scale evaluation by Berenzweig et al. [BLEW04] which not only examined audio-based descriptors but also subjective findings. As acoustic features they used MFCCs combined with GMM and neural networks. For subjective measures they used surveys, expert opinions, meta-data from AllMusic.com, playlist co-occurrences, personal user collections and web-text as source. One of the findings of the study was that computer-performed audio classification achieves agreement with ground-truth data that is at least comparable to the internal agreement between different subjective sources. The analysis also showed that the subjective measures from diverse sources show reasonable agreement, with the measure derived from co-occurrence in personal music collections being the most reliable overall. The collected data, particularly meta-data and MFCC features for more than 8700 tracks from 400 artists, has been made available online for future studies. In fact, part of this data set has been used also in the international MIREX evaluation campaign.

Goto et al. created a set of copyright-cleared music databases for research purposes, the RWC (Real World Computing) database [GHNO02, GHNO03]. It comprises five collections: popular music (english and japanese), classical music, jazz music, a music genre database containing 100 pieces from 10 genres and a musical instrument database[2].

Another benchmark data set for music classification (and clustering) is provided by Homburg et al. [HMM$^+$05] and contains 10-second excerpts of 1886 songs from 9 genres.

Stephen Downie outlines in [Dow03b] the way toward scientific evaluation of Music Information Retrieval systems. The paper describes the efforts that have been made and the discussions open for the construction and implementation of scientifically valid evaluation frameworks in the MIR research community. Specific focus was laid on the problematic topic of ground-truth assembly regarding "real-world" requirements and the development of a secure, but accessible, research environment that allows researchers to remotely access a large-scale test collection.

In 2005 the first Music Information Retrieval Evaluation eXchange (MIREX)[3] has been conducted by Stephen Downie's IMIRSEL team. The kick-off for scientific evaluation of MIR research was done one year earlier with the ISMIR 2004 Audio Description Contest[4].

2.5 Clustering, Visualization and Interfaces

Tzanetakis and Cook introduce a set of tools based on interactive 3D graphics for working with sound collections [TC00a]. The tools include sound analysis visualization displays (Timbregram, TimbreSpace, GenreGram) and model-based controllers for sound synthesis. Later, new tools for graphical query user interfaces were added creating a new paradigm for querying and browsing large audio collections [TEC02a].

Cano et al. report about applications of the FastMap algorithm for visualization of audio similarity and improved browsing of music archives [CKGB02]. Torrens et. al [THA04] present new interfaces for exploring personal music libraries in form of disc- and tree-map-based visualizations based on meta-data. A novel interface particularly developed for hand-held

[2] http://staff.aist.go.jp/m.goto/RWC-MDB/

[3] http://www.music-ir.org/mirex2005/

[4] http://ismir2004.ismir.net/ISMIR_Contest.html

devices has been presented by Gulik et al. [vGVvdW04]. This artist map interface clusters pieces of audio based on content features as well as metadata attributes using a a spring model algorithm.

Dixon shows the live tracking of musical performances with the "Performance Worm" using on-line time warping [Dix05]. The "Performance Worm" is an animation of some of the expressive parameters of a musical performance.

Goto presents a new interface to music collections called Musicream [GG05], in which pieces of music are represented by discs, that stream down on the screen enabling unexpected encounters with music pieces. Disc colors indicate the "mood" of a piece and thus reflect similarity in musical pieces. A "sticking function" attracts similar pieces like a magnet. The "metaplaylist function" enables advanced visual playlist arrangement with a high degree of freedom.

In recent years, Self-Organizing Maps have become very popular for the visualization of music collections. A Self-Organizing Map (SOM) is an unsupervised neural network providing a topology-preserving mapping from a high-dimensional input space onto a two-dimensional output space [Koh01]. The earliest works that use SOMs to organize sounds, based on pitch, duration and loudness, date back to [CPL94, FG94]. In [SP01] MFCCs are used for retrieval of sound events from a SOM.

Automatic organization of music collections on SOMs has been first demonstrated by Rauber et al. in [RF01], and later in [RPM02, PRM02, NDR05]. Pampalk presented an "Islands of Music" visualization using Self-Organizing Maps and Rhythm Pattern features [Pam01]. An interactive implementation of Islands of Music on both personal computers as well as portable devices has been shown by Neumayer et al. [NDR05].

Another work on exploring music collections by Pampalk et al. [PDW04] uses Aligned-SOMs, which allow for interactively changing the focus of organization among different aspects, like e.g. timbre or rhythm. Knees et al. [KPW04, PFW05] apply SOMs to organize music at the artist level using artist information mined from the web. Dittenbach et al. [DMR00] introduce the Growing Hierarchical Self-Organizing Map which extends the SOM principle to several layers and also enables a SOM to iteratively grow if data is sufficiently dense.

In [MUNS05] Mörchen et al. employ so-called Emergent SOMs for visualization of music collections, which are particularly suitable for creating large maps. Mayer et al. present another extension to SOMs, the Mnemonic SOM [MMR05] which allows the maps to take any arbitrary shape, for better memorization of locations on a SOM.

Other visualizations of music archives and music relations are emerging in the Internet, e.g. LivePlasma (formerly MusicPlasma)[5].

A review of visualization in audio-based Music Information Retrieval is provided in [CFPT06].

2.6 Conclusions

This chapter presented an overview of seminal works of the Music Information Retrieval research domain, which are relevant for the remaining chapters of this thesis. We have reviewed publications about content-based audio feature extraction, covering a large range of different features that can be extracted from music. Particularly relevant for this thesis are the various approaches for sound and music classification as well as the efforts for standard scientific benchmarking. Chapter 5 describes how music classification approaches are used both for evaluation of feature sets and classifiers as well as in scientific benchmarking campaigns.

Furthermore, we have revisited a number of graphical representations for music archives as well as interactive applications. The SOM-based approaches are particularly relevant for Chapter 6, which describes applications that are built predominantly on the Self-Organizing Map.

[5] `http://www.liveplasma.com/`

Chapter 3

Audio Feature Extraction

3.1 Introduction

For most Music Information Retrieval tasks music needs to be described in some way. As computers are not capable to grasp musical aspects directly, algorithms have been devised that extract features from music which are intended to capture semantics in music and to provide the basis for subsequent MIR tasks such as retrieval by similarity. As music can be stored in different representations, there are also multiple directions for the extraction of descriptors: Symbolic representations (e.g. MIDI files) provide directly the musical structure, such as note beginnings and pitch information, which can be used directly as part of a set of features. However, information about the sound of the music is completely lacking. Feature extraction from symbolic notations is not part of the topic of this thesis and is thus not considered in this chapter. On the other hand, audio-based approaches have to rely entirely on a mixed audio-signal. From the amplitude information it is very difficult to extract semantics and thus the majority of the audio-based algorithms perform transformations of the signal into the frequency domain, i.e. a spectrum analysis. From the energy and fluctuations of the individual frequency bands many aspects of the music can be derived, such as for instance pitch and rhythm information.

This chapter provides a brief overview of some of the many audio features developed and utilized in the MIR domain, starting with temporal and spectral low-level audio features such as energy and Spectral Centroid, continuing with the low-level audio descriptors defined in the MPEG-7 standard. Subsequently, MFCC features are described, which have originated in speech processing. A large set of features available in the MARSYAS system are described, including Fourier transform based features, MPEG compression based features, Wavelet transform features as well as features based on Beat and Pitch Histograms. Furthermore, Rhythm Patterns and two new feature sets, Statistical Spectrum Descriptors and Rhythm Histograms, are presented, which are the feature sets utilized in several studies, evaluations and benchmarking events (see Chapter 5) as well as in clustering applications for interaction with music collections (see Chapter 6).

3.2 Audio Features

3.2.1 Low-Level Audio Features

The following features are common low-level features employed in the context of many content-based audio retrieval projects, typically in combination with other features or feature sets. They are for example also available in audio software frameworks such as MARSYAS [Tza02], M2K [DEH05] or CLAM [AAG06].

Zero Crossing Rate

The Zero Crossing Rate (ZCR) is one of the features calculated directly from the audio wave form, i.e. in the time domain. It represents the number of times the signal crosses the 0-line, i.e. the signal changes from a positive to a negative value, within one second. It can be either a measure for the dominant frequency or the noisiness of a signal, and serves as a basic separator of speech and music.

RMS Energy

Root Mean Square (RMS) energy is computed in time domain by computing the mean of the square of all sample values in a time frame and taking the square root. Hence, it is a feature easy to implement. The RMS gives a good indication of loudness in a time frame and may also serve for higher-level tasks such as audio event detection, segmentation or tempo/beat estimation.

Low Energy Rate

Low Energy Rate is usually defined as the percentage of frames containing less energy than the average energy of all frames in a piece of audio. Energy is computed in time domain as RMS energy (see above). In [SS97] a frame is considered a low-energy-frame when it has less than 50 % of the average value within a one-second window.

Spectral Flux

Spectral Flux is a frequency domain feature and is computed as the squared differences in frequency distribution of two successive time frames. It measures the rate of local change in the spectrum. If there is much spectral change between two frames the Spectral Flux is high.

Spectral Centroid

The Spectral Centroid is the center of gravity, i.e. the balancing point of the spectrum. It is the frequency where the energy of all frequencies below that frequency is equal to the energy of all frequencies above that frequency and is a measure of brightness and general spectral shape.

Spectral Rolloff

Another measure of spectral shape is the Spectral Rolloff which is the 90 percentile of the spectral distribution. It is a measure of the skewness of the spectral shape.

3.2.2 MPEG-7 Audio Descriptors

The Moving Picture Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of standards for digitally coded representation of audio and video[1] . Until now, the group has produced several standards: The MPEG-1 standard is used e.g. for Video CDs and also defines several layers for audio compression, one of which (layer 3) is the very popular MP3 format. The MPEG-2 standard is another standard for video and audio compression and is used e.g. in DVDs and digital TV broadcasting.

[1] `http://www.chiariglione.org/mpeg/`

MPEG-4 is a standard for multimedia for the fixed and mobile web. MPEG-7 defines the Multimedia Content Description Interface and is the standard for description and search of audio and visual content. MPEG-21 defines the Multimedia Framework.

The MPEG-7 standard, part 4 [MPE02], describes a number of low-level audio descriptors as well as some high-level description tools. The five defined sets for high-level audio description are partly based on the low-level descriptors and are intended for specific applications (description of audio signature, instrument timbre, melody, spoken content as well as for general sound recognition and indexing) and will not be further considered here.

The low-level audio descriptors comprise 17 temporal and spectral descriptors, divided into seven classes. Some of them are based on basic waveform or spectral information while others use harmonic or timbral information. The following review of the 17 descriptors is based on an MPEG-7 overview provided by the ISO Organization on the web[2]:

Basic Temporal Descriptors

The two basic audio descriptors are temporally sampled scalar values for general use, also in combination with other low-level features.
The **AudioWaveform** Descriptor describes the audio waveform envelope (minimum and maximum), and is rather intended for display purposes.
The **AudioPower** Descriptor is similar to RMS energy and describes the power at certain intervals. It can be useful as a quick summary of a signal.

Basic Spectral Descriptors

The Basic Spectral Descriptors are derived from the signal transformed into the frequency domain, similar to the spectral low-level features described in Section 3.2.1. However, instead of an equi-spaced frequency spectrum, a logarithmic frequency spectrum is used, where the resulting frequency bins are spaced by a power-of-two divisor or a multiple of an octave. This logarithmic spectrum is the common basis for the four MPEG-7 basic spectral audio descriptors:
The **AudioSpectrumEnvelope** Descriptor computes the short-term power spectrum using the logarithmic frequency division and constitutes the evo-

[2] `http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm`

lution of the spectrum over time, hence a log-frequency spectrogram. The AudioSpectrumEnvelope can be used as a general-purpose descriptor for search and comparison or also for display purposes or even for a re-synthesis for an auralization of the data.

The **AudioSpectrumCentroid** Descriptor represents the center of gravity of the log-frequency power spectrum. Hence, it is a description of the shape of the power spectrum by a single scalar, indicating whether the spectral content of a signal is dominated by high or low frequencies.

The **AudioSpectrumSpread** Descriptor describes the second moment of the log-frequency power spectrum, indicating whether the power spectrum is centered near the spectral centroid, or spread out over the spectrum. This potentially enables discriminating between pure-tone and noise-like sounds.

The **AudioSpectrumFlatness** Descriptor describes the flatness of the spectrum of an audio signal for each of a number of frequency bands and is computed as the deviation of the power amplitude spectrum of each frame from a flat line. This descriptor may signal the presence of tonal components, if there is a high deviation from a flat spectral shape for a given band.

Signal Parameters

The two signal parameter descriptors estimate parameters of the signals which are fundamental for the extraction of other descriptors. The extraction of both of the following descriptors is possible however only mainly from periodic or quasi-periodic signals.

The **AudioFundamentalFrequency** Descriptor is intended to provide fundamental frequency of an audio signal. The MPEG-7 standard, however, does not give a reference implementation, thus, a number of different approaches could be taken for the determination of the fundamental frequency of a signal, such as pitch tracking for example. Consequently, the representation of this descriptor allows to include a confidence measure, in recognition of the facts that the various different extraction algorithms are not perfectly accurate and that there may be sections of a signal for which no fundamental frequency may be detected (e.g. noise).

The **AudioHarmonicity** Descriptor represents the harmonicity of a signal, allowing distinction between sounds with a harmonic spectrum (e.g. musical tones or voiced speech, such as vowels), sounds with an inharmonic spectrum (e.g. metallic or bell-like sounds) and sounds with a non-harmonic spectrum (e.g., noise, unvoiced speech, or dense mixtures of instruments). As for the AudioFundamentalFrequency Descriptor a concrete algorithm for describing AudioHarmonicity is not provided.

Timbral Temporal Descriptors

The **LogAttackTime** Descriptor characterizes the "attack" of a sound, i.e. the time it takes for the signal to rise from silence to the maximum amplitude.

The **TemporalCentroid** Descriptor also characterizes the signal envelope, representing the point in time that is the center of gravity of the energy of a signal. This descriptor may, for example, distinguish between a decaying piano note and a sustained organ note, when the lengths and the attacks of the two notes are identical.

Timbral Spectral Descriptors

The five timbral spectral descriptors are spectral features computed from a *linear-frequency* spectrum and are especially intended to capture musical timbre. The four harmonic spectral descriptors are derived from the components of harmonic peaks in the signal. Therefore, harmonic peak detection must be performed prior to feature extraction.

The **SpectralCentroid** Descriptor is determined by the frequency bin where the energy in the linear spectrum is balanced, i.e. half of the energy is below that frequency and half of the energy is above it. This is equal to the SpectralCentroid explained in Section 3.2.1 and differs from the MPEG-7 AudioSpectrumCentroid Descriptor by using a linear instead of a log-scale spectrum. It is included in the MPEG-7 standard for better distinguishing musical instrument timbres because is related to the perceptual feature of the "sharpness" of a sound.

The **HarmonicSpectralCentroid** is the amplitude-weighted mean of the harmonic peaks of the spectrum. It has a similar semantic as the other centroid Descriptors, but applies only to the harmonic parts of the musical tone.

The **HarmonicSpectralDeviation** Descriptor indicates the spectral deviation of logarithmic amplitude components from a global spectral envelope.

The **HarmonicSpectralSpread** describes the amplitude-weighted standard deviation of the harmonic peaks of the spectrum, normalized by the HarmonicSpectralCentroid.
The **HarmonicSpectralVariation** Descriptor is the correlation of the amplitude of the harmonic peaks between two sequential time frames of the signal, normalized by the HarmonicSpectralCentroid.

Spectral Basis Descriptors

The two spectral basis descriptors represent projections of high-dimensional descriptors to low-dimensional space for more compactness, which is useful e.g. for subsequent classification or indexing tasks.

The **AudioSpectrumBasis** Descriptor is a series of (potentially time-varying and/or statistically independent) basis functions that are derived from the singular value decomposition of a normalized power spectrum.

The **AudioSpectrumProjection** Descriptor is used together with the AudioSpectrumBasis Descriptor, and represents low-dimensional features of a spectrum after projection upon a reduced rank basis.

Silence Descriptor

The **Silence** Descriptor detects silent parts in audio and attaches this semantic to an audio segment. It may be used to aid segmentation of the audio stream or as a hint to not process a segment.

Pohle's work [Poh05] contains a slightly more detailed review of the MPEG-7 low-level audio descriptors as well as plots of several of the descriptors for exemplary songs from a variety of different genres. His thesis also includes a review of other research works using and evaluating MPEG-7 descriptors.

3.2.3 MFCCs

Mel Frequency Cepstral Coefficients (MFCCs) originated in research for speech processing and soon gained popularity in the field of music information retrieval [Log00]. A cepstrum is defined as the inverse Fourier transform of the logarithm of the spectrum. If the Mel scale is applied to the logarithmic spectrum before applying the inverse Fourier transform the result is called Mel Frequency Cepstral Coefficients.

The Mel scale is a perceptual scale found empirically through human listening tests and models perceived pitch distances. The reference point is 1000 Mels, equating a 1000 Hz tone, 40 dB above the hearing threshold. With increasing frequency, the intervals in Hz which produce equal increments in perceived pitch are getting larger and larger. Thus, the Mel scale is approximately a logarithmic scale, which corresponds more closely to the human auditory system than the linearly spaced frequency bands of a spectrum. In MFCC calculation often the Discrete Cosine Transform (DCT) is used instead of the inverse Fourier transform for practical reasons. From the MFCCs commonly only the first few (for instance 5 to 20) coefficients are used as features.

3.2.4 MARSYAS Features

The MARSYAS system is a software framework for audio analysis, feature extraction, synthesis and retrieval and contains a number of extractors for the following feature sets:

STFT-Spectrum based Features

MARSYAS implements the standard temporal and spectral low-level features described in Section 3.2.1: Spectral Centroid, Spectral Rolloff, Spectral Flux, RMS energy and Zero Crossings. Also, MFCC feature extraction is provided, c.f. Section 3.2.3.

MPEG Compression based Features

George Tzanetakis presented an approach which extracts audio features directly from MPEG compressed audio data (e.g. from mp3 files) [TC00b]. The idea was that in MPEG compression much of analysis is done already in the encoding stage, including a time-frequency analysis. The spectrum is divided into 32 equally spaced sub-bands via an analysis filterbank. Instead of decoding the information and again computing the spectrum this approach computes features directly from the 32 sub-bands in the MPEG data. Consequently, the derived features are called MPEG Centroid, MPEG Rolloff, MPEG Spectral Flux and MPEG RMS, and are computed similar as their non-MPEG counterparts (c.f. Section 3.2.1). These features should not be confused with the MPEG-7 standard features, which are based on *logarithmically* spaced spectrum data, whereas the MPEG-1 audio compression uses 32 *equally* spaced frequency bands.

Wavelet Transform Features

The Wavelet Transform [Mal99] is an alternative to the Fourier Transform which overcomes the issue of the trade-off between time and frequency resolution. For high frequency ranges, it provides low frequency resolution but high time resolution, whereas in low frequency ranges, it provides high frequency and lower time resolution. This is a closer representation of what the human ear perceives from sound.

The Wavelet Transform Features represent "sound texture" by applying the Wavelet Transform and computing statistics over the wavelet coefficients:

- mean of the absolute value of the coefficients in each frequency band
- standard deviation of the coefficients in each frequency band
- ratios of the mean absolute values between adjacent bands

These features provide information about the frequency distribution of the signal and its evolution over time.

Beat Histograms

The calculation of this set of features includes a beat detection algorithm which uses a Wavelet Transform to decompose the signal into octave frequency bands followed by envelope extraction and periodicity detection. The time domain amplitude envelope of each band is extracted separately which is achieved by full-wave rectification[3], low-pass filtering and downsampling. These envelopes are the summed together after removing the mean of each band signal and the autocorrelation of the resulting envelope is computed. The amplitude values of the dominant peaks of the autocorrelation function are then accumulated over the whole song into a Beat Histogram. This representation does not only capture the dominant beat in a sound, like other

[3] Full-wave rectification in the digital world means that each sample value is transformed into its absolute value.

automatic beat detectors, but captures more detailed information about the rhythmic content of a piece of music. The following set of features is derived from a Beat Histogram:

- relative amplitude (divided by the sum of amplitudes) of the first and second histogram peak
- ratio of the amplitude of the second peak to the amplitude of the first peak
- period of the first and second beat (in beats per minute)
- overall sum of the histogram, as indication of beat strength

Pitch Histograms

For the pitch content features, the multiple pitch detection algorithm described by [TK00] is utilized. The signal is decomposed into two frequency bands (below and above 1000 Hz) and amplitude envelopes are extracted for each of them using half-wave rectification[4] and low-pass filtering. The envelopes are then summed up and an enhanced autocorrelation function is used – similar as for Beat Histograms, but within smaller time frames (about 23 ms) – to detect the main pitches of the short sound segment. The three dominant peaks are then accumulated into a Pitch Histogram over the whole sound file. Each bin in the histogram corresponds to a musical note. Subsequently, also a folded version of the Pitch Histogram can be created by mapping the notes of all octaves onto a single octave. The unfolded version contains information about the pitch range of a piece of music and the folded version contains information about the pitch classes or the harmonic content. The following features are derived from Pitch Histograms:

- amplitude of the maximum peak of the folded histogram (i.e. magnitude of the most dominant pitch class)
- period of the maximum peak of the unfolded histogram (i.e. octave range of the dominant pitch)

[4] Half-wave rectification in the digital world means that each sample value < 0 is converted to 0.

- period of the maximum peak of the folded histogram (i.e. main pitch class)
- pitch interval between the two most prominent peaks of the folded histogram (i.e. main tonal interval relation)
- overall sum of the histogram (i.e. measure of strength of pitch detection)

MARSYAS can be applied to extract individual feature sets, however a set of combinations of them is defined and has been applied successfully in music genre classification. It also consists of additional tools for classification and subsequent processing. For detailed descriptions of all the features available in MARSYAS refer to [Tza02].

3.2.5 Rhythm Patterns

A Rhythm Pattern [Pam01, RPM02, RPM03], also called Fluctuation Pattern, is a matrix representation of fluctuations on critical bands. Parts of it describe rhythm in the narrow sense. The algorithm for extracting a Rhythm Pattern is a two stage process: First, from the spectral data the specific loudness sensation in Sone is computed for critical frequency bands. Second, the critical band scale Sonogram is transformed into a time-invariant domain resulting in a representation of modulation amplitudes per modulation frequency. The block diagram for the entire approach of Rhythm Patterns extraction is provided in Figure 3.2, steps of the first part are denoted with an ‘S’ and steps of the second part with an ‘R’.

In a pre-processing step the audio signal is converted to a mono signal (if necessary) and segmented into chunks of approximately 6 seconds[5]. Usually not every segment is used for audio feature extraction, the selection of segments however depends on the particular task. For music with a typical duration of about 4 minutes, frequently the first and last one or two (up to four) segments are skipped and from the remaining segments every third one is processed.

For each segment the spectrogram of the audio is computed using the short time Fast Fourier Transform (STFT). The window size is set to 23 ms[6]

[5] The segment size is 2^{18} samples with a sampling frequency of 44 kHz, 2^{17} for 22 kHz, and 2^{16} for 11 kHz, i.e. about 5.9 seconds.

[6] 1024 samples at 44 kHz, 512 samples at 22 kHz, 256 samples at 11 kHz

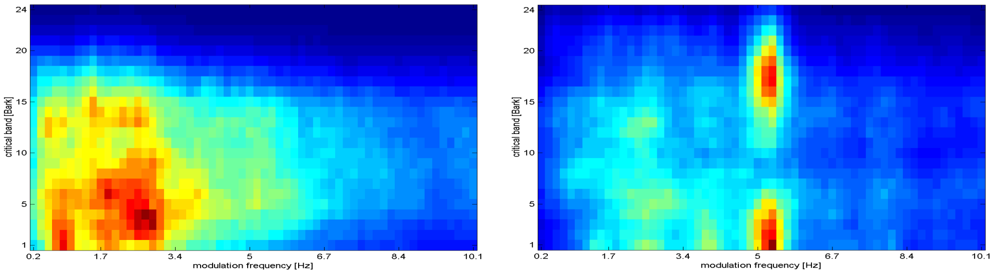
and a Hanning window is applied using 50 % overlap between the windows.

The Bark scale, a perceptual scale which groups frequencies to critical bands according to perceptive pitch regions [ZF99], is applied to the spectrogram, aggregating it to 24 frequency bands. A Spectral Masking spreading function is applied to the signal [SAH79], which models the occlusion of one sound by another sound.

The Bark scale spectrogram is then transformed into the decibel scale. Further psycho-acoustic transformations are applied: Computation of the Phon scale incorporates equal loudness curves, which account for the different perception of loudness at different frequencies [ZF99]. Subsequently, the values are transformed into the unit Sone, reflecting the specific loudness sensation of the human auditory system. The Sone scale relates to the Phon scale in the way that a doubling on the Sone scale sounds to the human ear like a doubling of the loudness.

In the second part, the varying energy on a critical band of the Bark scale Sonogram is regarded as a modulation of the amplitude over time. Using a Fourier Transform, the spectrum of this modulation signal is retrieved. In contrast to the time-dependent spectrogram data the result is now a time-invariant signal that contains magnitudes of modulation per modulation frequency per critical band. The occurrence of high amplitudes at the modulation frequency of 2 Hz on several critical bands for example indicates a rhythm at 120 beats per minute. The notion of rhythm ends above 15 Hz, where the sensation of roughness starts and goes up to 150 Hz, the limit where only three separately audible tones are perceivable. For the Rhythm Patterns feature set usually only information up to a modulation frequency of 10 Hz is considered. Subsequent to the Fourier Transform, modulation amplitudes are weighted according to a function of human sensation depending on modulation frequency, accentuating values around 4 Hz. The application of a gradient filter and Gaussian smoothing may improve similarity of Rhythm Patterns which is useful in classification and retrieval tasks.

The impact of this filtering and smoothing as well as of all of the psycho-acoustic transformations has been evaluated through experiments in this thesis, which are described in Chapter 5. The smoothing step seems not to be appropriate for all kinds of music collections and especially the Spectral Masking spreading function seems to introduce problems rather than being



(a) Classical: Johann Strauß

(b) Rock: Queens Of The Stone Age

Figure 3.1: Rhythm Patterns

beneficial. In contrast, the psycho-acoustically motivated transformations into the decibel, Phon and Sone scales have been identified as being crucial for audio feature extraction. The findings of these experiments led to an improved version of the Rhythm Patterns feature set, which has also been used in joint scientific MIREX evaluations.

A Rhythm Pattern is usually extracted per segment (e.g. 6 seconds) and the feature set is computed as the median of multiple Rhythm Patterns of a piece of music. The dimension of the feature set is 1440, if the full range of frequency bands (24) and modulation frequencies up to 10 Hz (60 bins at a resolution of 0.17 Hz) are used.

Figure 3.1 shows examples of Rhythm Patterns of a classical piece, the “Blue Danube Waltz” by Johann Strauß[7], and a rock piece, “Go With The Flow” by The Queens Of The Stone Age. While the rock piece shows a prominent rhythm at a modulation frequency of 5.34 Hz, both in the lower critical bands (bass) as well as in higher regions (percussion, e-guitars), the classical piece does not show a distinctive rhythm but contains a “blobby” area in the region of lower critical bands and low modulation frequencies. This is a typical indication of classical music.

3.2.6 Statistical Spectrum Descriptors

Statistical Spectrum Descriptors (SSD) [LR05] are based on the first part of the Rhythm Patterns algorithm, namely the computation of a psycho-acoustically motivated Bark scale Sonogram. However, instead of creating a

[7] Johann Strauß – An der schönen blauen Donau (op. 314), available free from `http://www.wien.gv.at/english/views/download/index.htm`, thanks to the municipality of Vienna and the Vienna Symphonic Orchestra.

pattern of modulation frequencies, an SSD intends to describe fluctuations on the critical frequency bands in a more compact representation, by deriving several statistical moments from each critical band. A block diagram of SSD computation is given in Figure 3.2.

The specific loudness sensation on different frequency bands is computed analogously to Rhythm Patterns (c.f. Section 3.2.5): A Short Time FFT is used to compute the spectrum. The resulting frequency bands are grouped to 24 critical bands, according to the Bark scale. Optionally, a spreading function is applied in order to account for spectral masking effects. Successively, the Bark scale spectrogram is transformed into the decibel, Phon and Sone scales. This results in a power spectrum that reflects human loudness sensation – a Bark scale Sonogram.

From this representation of perceived loudness a number of statistical moments is computed per critical band, in order to describe fluctuations within the critical bands extensively. Mean, median, variance, skewness, kurtosis, min- and max-value are computed for each band, and a Statistical Spectrum Descriptor is extracted for each selected segment. The SSD feature vector for a piece of audio is then calculated as either the mean or the median of the descriptors of its segments.

Statistical Spectrum Descriptors are able to capture additional timbral information compared to Rhythm Patterns, yet at a much lower dimension of the feature space (168 dimensions). Evaluations described in Chapter 5 show that SSD features are able to outperform RP features in music genre classification tasks.

3.2.7 Rhythm Histograms

Rhythm Histogram features are a descriptor for general rhythmic characteristics in a piece of audio. A modulation amplitude spectrum for critical bands according to the Bark scale is calculated, equally as for Rhythm Patterns (see Section 3.2.5 and Figure 3.2). Subsequently, the magnitudes of each modulation frequency bin of all 24 critical bands are summed up, to form a histogram of "rhythmic energy" per modulation frequency. The histogram contains 60 bins which reflect modulation frequency between 0.17 and 10 Hz[8] (c.f. Figure 3.3). For a given piece of audio, the Rhythm His-

[8] Using the parameters given in footnotes 5 and 6, the resolution of modulation frequencies is 0.17 Hz.

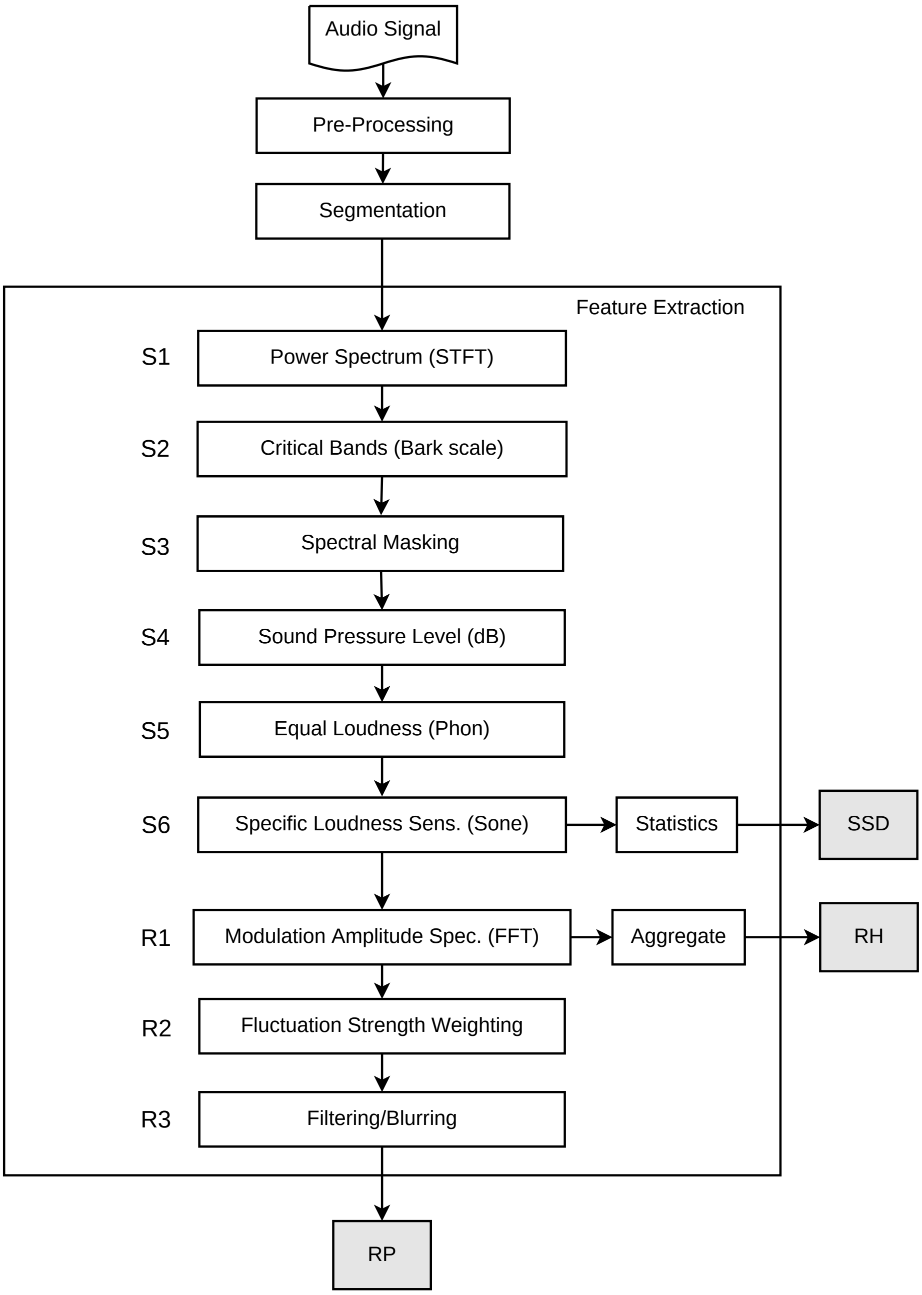
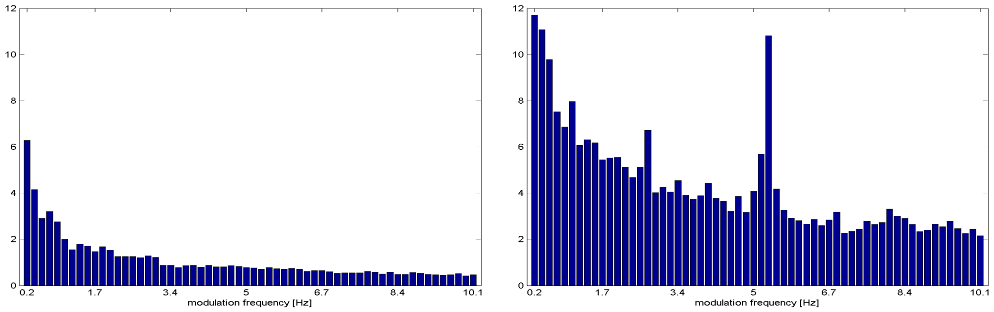


Figure 3.2: Feature extraction process for Statistical Spectrum Descriptors (SSD), Rhythm Histograms (RH) and Rhythm Patterns (RP)



(a) Classical: Johann Strauß

(b) Rock: Queens Of The Stone Age

Figure 3.3: Rhythm Histograms

togram feature set is calculated by taking the median of the histograms of every 6 second segment processed. The resulting feature vector has a 60 dimensions.

The Rhythm Histograms are similar in their representation to the Beat Histograms introduced by Tzanetakis (c.f. Section 3.2.4), the approach however is different: the Beat Histogram approach uses envelope extraction and autocorrelation and accumulates the histogram from the peaks of the autocorrelation function.

Figure 3.3 compares the Rhythm Histograms of a classical piece and a rock piece (the same example songs as for illustrating Rhythm Patterns have been used). The rock piece indicates a clear peak at a modulation frequency of 5.34 Hz while the classical piece generally contains less energy, having most of it at low modulation frequencies.

3.3 Conclusions

This section presented a review on commonly utilized audio features. Very often employed temporal and spectral low-level features which are easy to implement and available in a number of software packages have been described, follow by a review of the MPEG-7 standard features. Additional features available in the MARSYAS software framework have been presented. Of the three further feature sets described, two have been devised by myself – the Rhythm Histograms and the Statistical Spectrum Descriptors – while the Rhythm Patterns have undergone significant improvements throughout my work for this thesis.

All three of them have been evaluated in a number of experiments as well as in joint scientific evaluation campaigns and have proved competitiveness with state-of-the-art feature sets. Both the experiments and their conclusions as well as the results of international benchmarking evaluations are described in the following chapter. Subsequently, in Chapter 6 applications which are based on these extracted features are presented.

Chapter 4

Audio Collections

4.1 Introduction

Reference audio collections are very important for evaluation and benchmarking (c.f. Chapter 5). Without the use of standard benchmark collections the comparison of evaluation results would be impossible. Consequently there is a need for annotated (i.e. class-labeled) audio databases, the so-called ground-truth for evaluations.

This chapter introduces the audio collections used in this thesis, either for own experiments or in joint scientific evaluation campaigns, or in both. Some of them are publicly available or shared among researchers, others are not, because they are either copyrighted or undisclosed because they will be re-used in future MIR benchmark evaluations.

4.2 Audio Collections for Evaluation and Benchmarking

Table 4.1 gives an overview of the audio collections which are described in this chapter. It lists the short name of each collection, the file format used for encoding the music, the number of classes and number of files (songs, pieces) in each collection, the file length used in the collection (either full songs or excerpts) and the total playing time of each collection. The collections differ significantly in several characteristics, e.g. in the size (number of pieces), in the number and the particular set of categories they use or in audio quality. The following sections will take a more detailed look at each of them.

Table 4.1: Overview of music collections utilized in evaluations throughout this thesis (file encoding, number of classes and number of files in each collection, typical file duration and total duration of the collection [hh:min]).[2]

Name of collection	encoding	cl.	# files	file duration	duration
GTZAN	au, 22 kHz, mono	10	1000	30 seconds	05:20
ISMIR 2004 Genre	mp3, 44 kHz, stereo	6	1458	full songs	18:14
ISMIR 2004 Rhythm	RealAudio	8	698	30 seconds	05:39
MIREX 2005 Magnatune	wav, 22 kHz, mono	10	510	full songs	n/a
MIREX 2005 USPOP	wav, 22 kHz, mono	6	474	full songs	n/a
MIREX 2006 USPOP/USCRAP	mp3, 22 kHz, mono	9	5000	full songs	n/a
Mozart Collection	mp3, 44 kHz, stereo	17	2442	full songs	62:32

4.2.1 GTZAN

The GTZAN audio collection was assembled by George Tzanetakis and used in his dissertation for experiments with MARSAYS on genre classification, among others [Tza02]. Lateron it was used also by other research groups for evaluation and several publications exist based on usage of this collection. Also, several experiments conducted in Chapter 5 make use of this collection. It consists of 1000 pieces of audio equi-distributed among 10 popular music genres (see Table 4.2). The list of genres contains several genres which are not easy to separate and thus poses a challenge to automatic music classification systems. The pieces are 30-second excerpts and have a sampling rate of 22 kHz. The original format was the uncompressed AU format.

4.2.2 ISMIR 2004 Genre

The ISMIRgenre collection is from the ISMIR 2004 Genre Classification contest (c.f. Section 5.4.2) and contains 1458 songs from Magnatune.com[3], a “royalty free” Internet music provider. The music on Magnatune.com is subject to the Creative Commons License, which allows free non-commercial usage. Magnatune organizes its music within 8 genres on its web page, however the genres “New Age” and “Others” were not considered when the ISMIRgenre collection was compiled. The remaining genres are listed in Table 4.3(a). The songs in the collection are unequally distributed over the 6

[2] The number of files in the MIREX 2005 collections include the testing files only (the number of training instances was not available).
In the MIREX 2006 collection there was a 10th “class” labeled ”Cover Song”; the 330 cover songs are included in the total of 5000 files.

[3] http://www.magnatune.com/

Table 4.2: GTZAN collection (genres and number of tracks per genre)

genre	# tracks
blues	100
classical	100
country	100
disco	100
hiphop	100
jazz	100
metal	100
pop	100
reggae	100
rock	100
total	**1000**

genres. Thus, this collection, though not containing well-known music from popular artists, can be considered as a "real-world" music collection. The compiled collection was available from the ISMIR 2004 Genre Classification contest web site[4] in 128 kbps, 44 kHz, stereo MP3 format.

4.2.3 ISMIR 2004 Rhythm

The ISMIRrhythm collection is the one used in the ISMIR 2004 Rhythm classification contest. The source of the collection was the web site BallroomDancers.com[5] and a list of URLs to the files available from the contest web site[6] allowed to download those songs in RealAudio format, which then had to be converted to 44 kHz stereo Wave files. The collection consists of of 698 30-second excerpts of 8 genres from Latin and ballroom dance music (see Table 4.3(b)). The challenge of this collection is to distinguish this very restricted set of music genres by detecting the appropriate rhythm.

4.2.4 MIREX 2005 Magnatune

The first collection of the MIREX 2005 Audio Genre Classification task was once again taken from Magnatune.com, as in ISMIR 2004 Genre Classification. This time, however, an extended set of 10 genres has been used, which

[4] `http://ismir2004.ismir.net/genre_contest/index.htm`

[5] `http://www.ballroomdancers.com/Music/style.asp`

[6] `http://www.iua.upf.es/mtg/ismir2004/contest/rhythmContest/`

Table 4.3: ISMIR 2004 Genre and Rhythm audio collections: standard benchmark collections also used in many other evaluations (genres and number of tracks per genre)

(a) ISMIRgenre

genre	# tracks
Classical	640
Electronic	229
Jazz & Blues	52
Metal & Punk	90
Rock & Pop	203
World	244
total	1458

(b) ISMIRrhythm

genre	# tracks
ChaChaCha	111
Jive	60
Quickstep	82
Rumba	98
Samba	86
SlowWaltz	110
Tango	86
VienneseWaltz	65
total	**698**

made the task more challenging. Jazz & Blues has been separated into two genres and the genres Ambient, New Age, Ethnic and Folk have been added while World music was removed.

This time, neither a training set nor the final test set have been released to the participants, in order to enable future evaluations on the same database. During MIREX 2005 Audio Genre Classification, 1005 files have been used for training and 510 for testing. Only the genre distribution of the test set was made public and therefore Table 4.4 contains only the genre counts for the test set. MIREX 2005 used a hierarchical genre organization for evaluation on this database. The taxonomy joined the pairs of Jazz & Blues, Rock & Punk, Folk & Ethnic as well as 3 ‘electronical’ genres to a super-genre, while classical music constituted a genre of its own. The genre hierarchy is depicted in Figure 4.1.

4.2.5 MIREX 2005 USPOP

The second collection of the MIREX 2005 Audio Genre Classification task was part of the USPOP 2002 collection. The USPOP 2002 data set was compiled for several studies by a team from Columbia University [BLEW04]. The original collection consists of 706 albums and 8764 tracks from 400

Table 4.4: MIREX 2005 collections used for Audio Genre Classification (genres and number of *test instances* per genre, the distribution of the training instances was not available)

(a) Magnatune

genre	# tracks
Ambient	34
Blues	34
Classical	79
Electronic	82
Ethnic	83
Folk	24
Jazz	22
New age	34
Punk	34
Rock	84
testing	**510**
training	**1005**
total	**1515**

(b) USPOP 2002

genre	# tracks
Country	84
Electronica & Dance	67
New Age	21
Rap & Hip-hop	117
Reggae	18
Rock	167
testing	**474**
training	**940**
total	**1414**

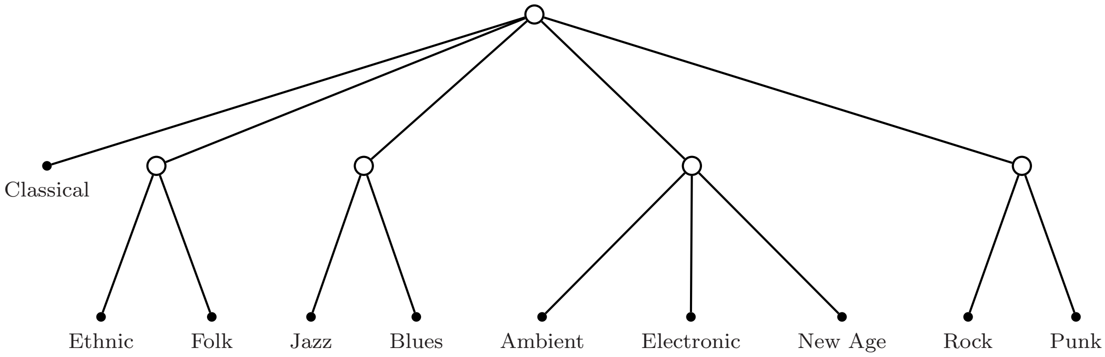


Figure 4.1: MIREX 2005 Magnatune genre hierarchy

artists[7]. The list of tracks can be obtained[8], however not the actual music, as it is copyrighted popular music (purchased by the Columbia team) and not freely distributable. The benefit of this is that this time "real-world" music was used for evaluation. Nevertheless, interested researchers can obtain MFCC features computed from the USPOP 2002 data set.

Genre assignments to the USPOP 2002 data set were made artist-wise by gathering the "style" tags from All Music Guide[9] for each of the 400 artists represented in the collection. Each of the artist was assigned multiple genre tags, the list of assignments can also be obtained[10].

In MIREX 2005 a subset of 1414 tracks of the USPOP 2002 collection has been selected. The IMIRSEL team purchased the CDs from this subset and compiled a set of 940 training files and 474 files for testing. Genre assignments were not adopted from the USPOP 2002 labels but from a public meta-data provider such as Gracenote or freedb.org.

The list of genres and number of test instances per genre is given in Table 4.4(b). Neither the list of tracks nor the audio data was released to the participants.

4.2.6 MIREX 2006 USPOP/USCRAP

The MIREX 2006 collection for Audio Music Similarity and Retrieval was again selected from the USPOP data set and included also music from the USCRAP collection and the Cover Song collection. The USCRAP collection was compiled and acquired by the IMIRSEL team and contains another set of US pop music. 30 cover songs with 11 versions each were included within this collection in order to accommodate the MIREX 2006 Cover Song Identification task. In total the collection comprised 5000 tracks from 9 genres (c.f. Table 4.5).

All tracks were handled in the exactly the same way: The CDs were encoded into MP3 with a variable bitrate of 192 kbps and 44.1 kHz sampling frequency and later decoded for the contest to 22 kHz mono WAV audio. It was considered that no track should be shorter than 30 seconds or longer than 10 minutes and that there was a maximum of 20 tracks per artist and

[7] `http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html`

[8] `http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002-aset.txt`

[9] `http://www.allmusic.com`

[10] `http://labrosa.ee.columbia.edu/projects/musicsim/aset400-styles.txt`

Table 4.5: MIREX 2006 music collection: music selected from both the USPOP and USCRAP collections, utilized for the Audio Music Similarity and Retrieval task (genres and number of tracks per genre; songs which were used for Cover Song Detection were labeled with ‘Cover Song’).

genre	# tracks
Country	246
Electronica & Dance	453
Jazz	87
Latin	62
New Age	69
R & B	82
Rap & Hip-Hop	1244
Reggae	93
Rock	2334
Cover Song	330
total	**5000**

a minimum of 50 tracks per labelled genre. This collection is the largest database used in joint scientific evaluation until now. However, neither the list of tracks nor the actual audio were made available.

4.2.7 Mozart Collection

This collection comprises the complete works of Wolfgang Amadeus Mozart. The collection was available on 170 CDs and was encoded to 256 kbps constant bitrate MP3 format. It originally consisted of 2443 pieces, one track was removed because it consisted of a spoken sentence shorter than 5 seconds only. The music has been divided into a set of categories, which were derived partly from the categorization of the CD collection (according to the CD covers) and partly by further subdivision of the main categories. There are 17 classes in total which are listed in Table 4.6. The total playing time of the music is 62 hours and 32 minutes.

The remarkable characteristic of this collection is that it was composed by a single composer and is hence a very homogeneous collection, consisting entirely of classical music from one specific period of time. It is thus a particular challenge for MIR algorithms to derive features for a proper organization of this collection.

Table 4.6: Mozart collection: the complete works of Wolfgang Amadeus Mozart (categories and number of pieces per category).

category	# pieces
Canons	41
Church Sonatas	17
Concert Arias	53
Concertos	159
Dances	207
Divertimenti	169
Flute Quartets & Sonatas	27
Horn, Oboe & Clarinet Ensembles	10
Keyboard Works	146
Operas	768
Piano Ensembles	30
Sacred Works	324
Serenades	77
Songs	33
String Ensembles	130
Symphonies	144
Violin Sonatas	107
total	**2442**

4.3 Conclusions

This chapter introduced several audio collections used for benchmarking campaigns or other evaluations. They are characterized by a different number of files, a different number of classes, different stratifications (i.e. genre distributions) and different homogeneity (different types of categories). Consequently, evaluation on many different audio databases will induce the generalization of the applied approaches to real world problems. Using these databases for evaluation allows to not only compare the performance of different feature sets from different research institutions. They also show the different performance of the same feature type across different tasks, highlighting the fact that no single one is optimal for all situations.

Chapter 5

Evaluation and Benchmarking

5.1 Introduction: History of Evaluation in MIR Research

The increased interest on research in the MIR domain and the growing number of approaches to different problems in MIR soon called for a standardized evaluation of the different methods proposed. The idea of a common scientific evaluation of MIR algorithms existed already at the time of the first ISMIR symposium in the year 2000 [Dow02]. At ISMIR 2001 discussion started about details such as evaluation frameworks, standardized test collections, tasks and evaluation metrics. The need of an evaluation framework for the growing MIR research community was expressed in a resolution signed by more than 90 researchers from the multidisciplinary and multinational MIR/MDL (music digital libraries) research community. A Workshop at JCDL 2002 brought the MIR/MDL people into contact with people from the Text REtrieval Conference (TREC), particularly Ellen Voorhees, who gave a keynote talk about the potential applicability of the TREC evaluation paradigm to the needs of the MIR/MDL community. TREC at that time was already a well-established forum with the aim to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies[1]. However, soon it was clear that the TREC evaluation methodologies are not

[1] http://trec.nist.gov/overview.html

directly applicable to the music domain and hence from thereon the term "TREC-like evaluation" was used for the planning of an MIR evaluation, in fact not related to TREC more than through inspiration and occasional inquiry. Among the particularities of MIR evaluation are the different representations of music (audio, symbolic representations, scores, metadata), the types of different queries and thus tasks (song retrieval, score following, beat detection, etc.) and the identification of real-world application scenarios (libraries/archives, educational, professional, personal use) [Dow03b]. In 2002 the IMIRSEL (International Music Information Retrieval Systems Evaluation Laboratory) project was started at the University of Illinois at Urbana-Champaign (UIUC) with J. Stephen Downie as the project principal investigator. IMIRSEL was started to bundle the ongoing efforts for the realization of a scientific evaluation, particularly the establishment of the necessary resources for the assembly of music databases and other necessary data, the compilation of evaluation tasks and the selection of appropriate evaluation metrics. At ISMIR 2002 a panel on MIR evaluation frameworks was held, stating the question "How do we move forward on making a TREC-like evaluation scenario for MIR/MDL a reality?" [Dow02]. It was clear that still much work had to be done, and apart from the issues described above, copyright issues for the compilation of a standardized, yet as much as possible "real-world", music database arised. The copyright issues for music information retrieval research is still a topic of ongoing discussions, obviated by two bypassing solutions: (1) use of copyright-free music or (2) evaluation in a central place with a secured music database of copyrighted music, with no access of the evaluation participants to the actual music. Intellectual property not only in terms of music authorship but also regarding "music information" that derives from MIR systems and the implied rights and liabilities were discussed in a panel at ISMIR 2003, next to another panel on "Making Music Information Retrieval Evaluation Scenarios a Reality", discussing about how to arrive at a community consensus about the specific tasks to be evaluated and the metrics to be used.

Much effort has been put from the organizers of ISMIR 2004 to realize the first "Audio Description Contest"[2]. This first evaluation of MIR algorithms allowed the submission of algorithms in 5 categories: Genre Classification, Artist Identification, Melody Extraction, Tempo Induction and

[2] http://ismir2004.ismir.net/ISMIR_Contest.html

Rhythm Classification. Knowing that these "contest" categories did not represent the full range of MIR developments, the organizers knew as well that this would be an important kick-off for future MIR evaluations. The 2004 evaluation was organized as a "contest", meaning that there was a winner in each of the categories, according to evaluation metrics that were agreed on beforehand among the participants. In this first evaluation also test data was made available to participants (as far as copyright licenses allowed), in order to assist participants in assembling their algorithms in a proper format for the evaluation. The evaluation took place at the labs of the ISMIR 2004 organizers, MTG at University of Pompeu Fabra, Barcelona, before the conference.

The second scientific evaluation of MIR algorithms was accomplished by the IMIRSEL project during and before ISMIR 2005 and was henceforth called MIREX (Music Information Retrieval Evaluation eXchange). In MIREX 2005 anyone who liked to participate could propose new tasks. Ten different tasks have been defined, for which people could submit their algorithms. Besides seven audio-based tasks also three tasks for symbolic music notations were available, hence a major extension of disciplines over the 2004 evaluation. Observing that the performance of a number of algorithms delivered results at a very similar level, it was decided that significance tests should be introduced for future evaluations.

As a further reaction to these similar results a (partly) changed set of tasks has been proposed for MIREX 2006: For instance, instead of a genre classification task, similarity algorithms this time were graded by human judgements, through the first human listening test within joint scientific MIR/MDL evaluation. Another new task was Audio Cover Song Identification. In total nine tasks were available at MIREX 2006, which were proposed, discussed and agreed upon beforehand by means of a mailing list[3] as well as a Wiki[4]. A significance test that was applied to several of the tasks showed that there were indeed no significant differences between the results of many of the algorithms, a fact that was already presumed in MIREX 2005.

Annual scientific evaluations of algorithms play an important role in research: Not only do they allow a comparison of the state-of-the-art in

[3] https://mail.lis.uiuc.edu/mailman/listinfo/evalfest

[4] http://www.music-ir.org/mirex2006

a particular field, they also enable research teams to measure their own individual progress over the years. In both aspects, evaluation not only supports research in MIR, it actively stimulates and fosters research, giving also new incentives and encouraging communication and exchange between the research teams. At this point it should be stated, that the evaluation forums are not limited to research groups, but are open to any organization or individual who wishes to participate with his or her own approach to a certain topic.

I have participated in each of the annual evaluations up to now, in several tasks. The following sections describe the evaluation tasks I participated in, the approaches I submitted and discusses the results in comparison to other participants of the evaluations. Section 5.2 explains common evaluation methodologies and presents typical measures used in evaluations. Section 5.3 starts with an outline of the situation when I started to work in the MIR domain, describes the efforts I made for improvements of existing approaches and outlines the experiments I did and the evaluations I was involved in. Section 5.4 describes the ISMIR 2004 Audio Description Contest, the first state-of-the-art MIR algorithm evaluation, and presents the results of the three tasks I participated in.

Section 5.5 explains the numerous experiments I did on the evaluation of psycho-acoustic transformations involved in audio feature extraction as well as further experiments on two newly developed feature sets. Furthermore it describes an approach of combining the feature sets in an effort to further improve classification results. The experiment results are then compared to other published results on de-facto standard evaluation benchmark databases.

Section 5.6 describes the tasks, approaches and results of the ISMIR 2005 evaluation forum, now called MIREX. In Section 5.7 the changed requirements for the 2006 evaluation round are described, as well as an evaluation of distance metric for similarity-based retrieval. Eventually, Section 5.8 presents the new tasks of MIREX 2006, outlines the first human listening test done for state-of-the-art MIR evaluations and describes the many evaluation metrics and results, closing with conclusions thereof.

Section 5.9 summarizes the evaluations described in this chapter and outlines their implications.

5.2 Evaluation Methods and Measures

5.2.1 Classification

Quantitative evaluations of features for Music Information Retrieval are typically performed through classification tasks. The task is to categorize pieces of music into a pre-determined list of classes, e.g. genres. The very active research domain of machine learning has developed numerous classifiers, which can be employed for music classification tasks. All of them intend to find a separation of classes within the feature space which is spanned by the feature vectors computed from music. Obviously that space has to be populated first with training data, i.e. classification is only possible if a part of the data is available already labeled with classes. The classifier can then learn from the labeled data and induce models for future items.

A simple classifier is the nearest neighbor classifier. It matches an unlabeled data item to the closest item of the labeled ones and induces the class prediction from the label of that item. A k-nearest neighbor classifier (k-NN) considers multiple (k) items and derives the prediction from the most frequent class label. Different distance metrics play an important role in k-NN classification (c.f. Section 5.7.2).

The Perceptron [Ros58] is an iterative classifier that starts with a randomly initialized hyperplane, which is updated as the feature vectors are presented in each iteration. The Perceptron algorithm has been shown to converge to an optimal solution with no mis-classification in case the data set is linearly separable. However, the Perceptron is not deterministic since it depends on its initialization, and the order that the samples are presented during training.

A Support Vector Machine (SVM, [Vap95]) is a classifier that constructs an optimal separating hyperplane between two classes. The hyperplane is computed by solving a quadratic programming optimization problem, such that the distance of the hyperplane from its closest data vectors is maximized. A “soft margin” allows a number of points to violate these boundaries. Except for linear SVMs the hyperplane is not constructed in feature space, but a kernel is used to project the feature vectors to a higher-dimensional space, in which the problem becomes linearly separable. Polynomial or radial basis function (RBF) kernels are common.

The linear SVM usually performs better in classification than other linear classifiers, especially for high-dimensional data sets. However, it is significantly more expensive to compute.

Many more classifiers exist. For an in-depth review of machine learning and pattern classification algorithms refer to [DHS00].

5.2.2 Cross-Validation

For evaluation through a classification task a data set in which all items (songs) are labeled by a class (genre) is needed. The labeled data set – the so-called ground-truth – is usually split into a training and a test set. The labeled training set is then used for training the classifier and building a model, and afterwards the test set (without the labels) is used to predict classes from the model. Subsequently the predictions are compared to the test set labels and an Accuracy value (among other measures) is computed from the result. Typically, a 2:1 training/test set split is used, i.e. 67 % of the music in the collection is used for training and 33 % for testing. If parameters have to be tuned for a particular classifier, also the constellation of training – development – test set is common. The development set is then used for selection (determination) of the best parameters, while the test set is left for final testing.

As the results usually vary depending on what part of the collection has been selected for training and what is the test data, the n-fold cross-validation approach has been introduced, which is supposed to give more stable results. In a 10-fold cross-validation for instance, the collection is split into 10 (random) equal-sized sub-sets. In 10 runs, each sub-set is once selected as test set, while the other 9 sub-sets are used for training. Measures are then calculated from each of the 10 tests and the final result is averaged. An n-fold cross-validation is referred to be "stratified" when each of the sub-sets contains the same class distribution as the entire data set.

In my classification experiments in the following sections, a stratified 10-fold cross-validation approach is used. The scientific evaluation campaigns usually use a training/test set split, due to time limitations.

5.2.3 Evaluation Measures

Feature sets and classification techniques are evaluated using a range of measures. The most commonly used one is Accuracy. In a two-class problem Accuracy is defined as

$$A = \frac{TP + TN}{N} \tag{5.1}$$

TP is the number of true positives, TN is the number of true negatives, i.e. the two cases where the classifier predicted the correct class label. A false negative (FN) is when the classifier prediction was 'false' while it should have been 'true'. A false positive (FP) appears, when the classifier assigns an item, that is actually labeled as 'false', to the class 'true'. N is the number of items in the collection.

In music classification usually more than two classes exist, thus Accuracy is computed as the sum of all correctly classified songs, divided by the total number of songs in a collection:

$$A = \frac{\sum_{i=1}^{|C|} TP_i}{N} \tag{5.2}$$

The determination of the number of correctly classified songs implies the availability of genre (class) labels for all songs in the collection, i.e. for evaluation a music collection with 100 % annotated data is needed, which is called ground-truth data.

Besides Accuracy, Precision and Recall are further performance measures often reported from classification tasks:

$$\pi_i = \frac{TP_i}{TP_i + FP_i}, \quad P^M = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \tag{5.3}$$

π_i is the Precision per class, where TP_i is the number of true positives in class i and FP_i is the number of false positives in class i, i.e. songs identified as class i but actually belonging to another class. $|C|$ is the number of classes in a collection and P^M is macro-averaged Precision. Macro-averaging computes the Precision per class first and then averages the Precision values over all classes, while micro-averaged Precision is computed by summing over all classes. In micro-averaging, however, large classes are over-emphasized and therefore only macro-averaging is used in this thesis. Precision measures the proportion of relevant pieces to all songs retrieved.

$$\rho_i = \frac{TP_i}{TP_i + FN_i}, \quad R^M = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|} \tag{5.4}$$

ρ_i is the Recall per class, where FN_i is the number of false negatives of class i, i.e. songs belonging to class i, but which the classifier assigned to another class. R^M is macro-averaged Recall, micro-averaged Recall is computed, analogously to micro-averaged Precision, directly by summing over all classes. Recall measures the proportion of relevant songs retrieved out of all relevant songs available.

An additional performance measure is the F-measure, which is a combined measure of Precision and Recall, computed as their weighted harmonic mean. The most common F-measure is F_1-measure, which weights Precision and Recall equally:

$$F_1 = \frac{2 \cdot P^M \cdot R^M}{P^M + R^M} \tag{5.5}$$

The general definition for the F-measure is

$$F_\alpha = \frac{(1+\alpha) \cdot P^M \cdot R^M}{\alpha \cdot P^M + R^M} \tag{5.6}$$

where α influences the weighting of Precision and Recall.

As the performance measures may fluctuate depending on the particular partitioning of the data collection into sub-sets used for training and testing, usually a cross-validation approach is chosen in order to get a more stable assessment of the classifier results.

5.3 Starting Point

When I started to work on MIR in 2003, one of the first works I did was a comparison of two audio similarity feature sets by clustering music on a Self-Organizing Map [Lid03]. I compared the MARSYAS Genre feature set (c.f. Section 3.2.4) and the Rhythm Patterns features (c.f. Section 3.2.5), using a small personal music collection with 335 audio files. MARSYAS 0.1 and Elias Pampalk's Music Analysis Toolbox for Matlab, version 0.2 from 2002 were used for feature extraction. In this evaluation the Rhythm Patterns features delivered better results than MARSYAS. I continued to use the MA Toolbox in my further work and successively optimized the Matlab code for the extraction of Rhythm Patterns from audio data. One of the modifications made was the introduction of many new options which enabled the control of which parts of the feature extraction process to be included or not. This allowed to evaluate the influence of several psycho-acoustic transformation steps. Other options offered automatic resampling of the audio data and allowed to control important parameters such as the Hanning window size, the number of critical frequency bands and the range of modulation frequencies to be considered in the Rhythm Patterns (see Section 3.2.5). Another important addition was to control the number of segments to be selected for the feature extraction process as well as the time to be skipped from the start and end of an audio file, in order to avoid lead-in and fade-out effects. I also made an effort to optimize the processing performance and analyzed the computational cost of each of the parts of the algorithm. It turned out that the conversion from the Decibel to the Phon scale, which incorporates numerous non-linear transformations and look-ups in tables, amounted about 71 % of the total processing time. After re-implementing the Phon conversion I reduced this portion to 3.2 % and achieved a reduction of the total feature extraction time of about 70 %. Furthermore, I introduced some checks to improve the robustness of the code. One particular check – whether the options set for segment selection and skipping of lead-in/fade-out match the actual duration of the audio file – made the algorithm pass the unannounced ISMIR 2004 robustness test (see Section 5.4.2). The large number of introduced options enabled a range of experiments, which lateron led to improvements in the algorithm itself (c.f. Sections 5.5 and 5.6). In 2005 I completely re-implemented the Rhythm Patterns audio feature extraction in Matlab in order to do more optimizations of the algorithm. Based on the Rhythm Patterns feature extractor I developed two new feature sets which were then available in the same Matlab package. Later, in 2006, the feature extractor was re-implemented again in Java in order to be included in interactive applications for browsing and retrieval of music archives.

5.4 ISMIR 2004 Audio Description Contest

In 2004 the first joint scientific MIR evaluation has been started as the "ISMIR 2004 Audio Description Contest"[5]. Five tasks were available:

- Genre Classification
- Artist Identification
- Melody Extraction
- Tempo Induction
- Rhythm Classification

I participated, together with Andreas Rauber and Andreas Pesenhofer in a team, in three of the tasks: Genre Classification, Artist Identification and Rhythm Classification. Details of the contest requirements, preparation of the submission, training data and, most importantly, evaluation results are discussed in the following subsections.

5.4.1 Submitted Algorithm and Contest Preparations

As the 2004 contest was the first evaluation within the MIR domain, there were no standardized submission formats or guidelines. Therefore, each of the tasks had its own rules of how to submit, what to submit and what to produce as output. Also the data given to the participants differed among the tasks: For Genre Classification a training set of complete audio files was provided to the participants beforehand in order to enable them to train their classifier models to music of the same genres as used in the final test database and to test their systems for conformance to the submission requirements. For Rhythm Classification there was also a training set available, however only as 30-seconds excerpts. For Artist Identification due to copyright issues only low-level (MFCC) features extracted beforehand by the contest organizers had been released (which we did not use).

The participants could experiment with that data and different classifiers and had to submit an entire system including the classifier and, for Rhythm Classification, also the trained model.

We used the optimized implementation of the Rhythm Patterns feature set in our submissions to the three tasks. The full implementation including

[5] http://ismir2004.ismir.net/ISMIR_Contest.html

all psycho-acoustic transformation steps (c.f. Section 3.2.5) had been used. Segmentation of the audio files was done into 5.9 seconds segments, the first and the last segment were skipped and every third segment of the remaining ones was processed. Rhythm Patterns were extracted on 24 critical bands and 60 modulation frequencies (with a resolution of 0.17 Hz).

Much effort had to be done to meet the requirements of the contest, because we did not have an entire audio classification system at that time. Thus, a number of scripts had to be written in order to provide an entire framework which would extract the features from audio, train a model, perform the classification and write correct output corresponding to the required format. While for the Genre and Artist Classification tasks a framework doing all these steps had to be submitted (i.e. the classifier model would be trained during evaluation) the requirements of the Rhythm Classification had foreseen, that a model had to be trained in advance, which would then be submitted together with the feature extraction algorithm and the classifier.

We did a range of experiments with different classifiers and measured the performance on both the training sets of the Genre and Rhythm Classification tasks. From among the different classification algorithms we tested, we chose Support Vector Machines (SVMs) as the one to use for the contests, as they outperformed all other classifiers in our experiments. We used linear SVMs in the SMO implementation of the Weka Machine Learning software[6] [WF05].

5.4.2 Genre Classification

Music from Magnatune[7] has been used in this task and the task was to classify the music into the same set of genres that Magnatune uses to organize the music on their web site[8]. In this contest a database of 1458 songs has been used, half of the songs were released beforehand to the participants while the other half was kept closed for evaluation. Table 4.3(a) shows the distribution of songs among the six genres, equal distribution existed in both the training and test set (50 % each). Besides the training data, a sample

[6] `http://www.cs.waikato.ac.nz/ml/weka/`

[7] `http://www.magnatune.com/` – Magnatune's licensing scheme allows the use of the music they publish for research.

[8] `http://ismir2004.ismir.net/genre_contest/index.htm`

Table 5.1: Results of the ISMIR 2004 Genre Classification contest: Overall Accuracy and Accuracy normalized by genre frequency (in %)

Participant	A	A (norm.)
Elias Pampalk	84.1	78.8
Kris West	78.3	67.2
George Tzanetakis	71.3	58.6
Thomas Lidy and Andreas Rauber	70.4	55.7
Dan Ellis and Brian Whitman	64.3	51.5

Table 5.2: Results of the unannounced robustness test of the ISMIR 2004 Genre Classification contest: Overall Accuracy and Accuracy normalized by genre frequency in % (other participants failed)

Participant	A	A (norm.)
Thomas Lidy and Andreas Rauber	63.4	52.1
George Tzanetakis	57.5	24.0

framework of scripts was provided to the participants to allow them to simulate the evaluation environment. We had to plug in a number of additional scripts in order to perform automatic classification of the features extracted from Matlab through the Weka Java software.

There were five participants in this task. Table 5.1 shows the evaluation results in terms of Accuracy (percentage of correctly classified tracks) and Accuracy normalized with respect to the probability of each class (i.e. average per-class Accuracy, which is equal to macro-averaged Recall as defined in Section 5.2.3)[9]. We achieved 70.4 % Accuracy on the Genre Classification and thus rank four.

Elias Pampalk, who won the contest, used Gaussian Mixture Models (with 30 Gaussians) and Expectation Maximization to cluster frame-level features and performed the Genre Classification using nearest neighbor classification on cluster similarity. 19 MFCC coefficients were extracted per frame and a piece of audio was summarized by clustering the frame-level features and thus finding a typical representation, variances and prior probabilities. For similarity (resp. distance) computation 2000 samples were

[9] `http://ismir2004.ismir.net/genre_contest/results.htm`

drawn from each Gaussian Mixture Model. This approach exceeded by far the maximum computation time that was introduced in later MIREX evaluations, i.e. it needed several days to compute.

During the realization of the Genre Classification contest an additional – unannounced – robustness test was performed on the five participating systems[10]: A 25 second excerpt was extracted from the middle of the audio files and the five algorithms were tested whether their performance would decrease in the case of using short audio excerpts. Three of the algorithms failed in extracting features from 25 seconds audio, only two succeeded (c.f. Table 5.2): Our algorithm performed the genre classification with 63.4 % Accuracy (i.e. a decrease of 7 percentage points over the full pieces of audio) and George Tzanetakis' achieved 57.5 % (a decrease of 13.8 percentage points, which means that it fell behind us in the ranking). This result would not have been achieved without the additional robustness I implemented into the algorithm prior to the contest.

5.4.3 Artist Identification

The Artist Identification contest used the same framework as the Genre Classification contest, but a different music database. The task was to identify artists given 3 songs per artist after training the system on 7 songs per artist[11]. For training and development two sets of low-level features were provided to the participants, corresponding to songs of 105 artists from the USPOP2002 collection [BLEW03]. The training set included 7 songs from each artist and the development set 3 songs. However, the features provided were MFCC features, which we could not use for our Rhythm Patterns based approach. Therefore we submitted our algorithm without prior training on Artist Identification and with the same parameters as for Genre Classification. The evaluation test set consisted of about 200 artists, which were *not* part of the USPOP2002 collection. During the evaluation the algorithms were given 7 songs for training, and 3 songs per artist were used for evaluation. According to the contest organizers, due to technical limitations, the original aim of testing the algorithms on all 200 artists could not be achieved and therefore the submissions were evaluated both with 30 and 40 artists[12].

[10] http://ismir2004.ismir.net/download/PanelAudioContest.pdf

[11] http://ismir2004.ismir.net/genre_contest/index.htm#artist

[12] http://ismir2004.ismir.net/genre_contest/results.htm#ArtistResult

Table 5.3: Results of the ISMIR 2004 Artist Identification contest: Accuracy in % on identifying 30 resp. 40 artists

Participant	30 artists	40 artists
Dan Ellis and Brian Whitman	34	24
Thomas Lidy and Andreas Rauber	28	24

Only two teams participated in this task. Table 5.3 presents the results of the Artist Identification contest. Dan Ellis and Brian Whitman achieved 34 % Accuracy on identifying 30 different artists, we achieved 28 %. For the increased problem of 40 artists, both systems delivered equal results of 24 % Accuracy.

This task can be compared to Genre Classification regarding it as a classification problem with an increased number of classes: While on the 6 class Genre Classification problem our approach achieved 55.7 % normalized Accuracy and Ellis and Whitman's 51 %, our result dropped to 28 % on the 30 class Artist Identification problem and theirs only to 34 %. Note that our system was *not* designed for Artist Identification and we did no prior experiments on that task, whereas Ellis and Whitman did specific research on Artist Similarity [EWBL02].

5.4.4 Rhythm Classification

The third task we participated within the ISMIR2004 Audio Description Contest was on Rhythm Classification. The task was to classify pieces of ballroom dance music correctly into 8 available classes[13]. A set of 488 training instances (30 seconds excerpts in RealAudio format from ballroom-dancers.com, which had to be decoded to wav format) was provided to the participants. These training instances had to be used to train a model with one's system before the contest. The contest required to submit this model together with one's algorithm, which means, that during the contest no training was made and the submitted algorithm had to extract features only from the test instances and to predict their genres. Table 4.3(b) contains the list of genres involved in the Rhythm Classification task as well as the number of instances per class in the music collection. A stratified 70:30

[13] http://www.iua.upf.es/mtg/ismir2004/contest/rhythmContest/

Table 5.4: Result of the ISMIR 2004 Rhythm Classification contest and comparison with other results on the same audio collection: Accuracy in %, $A*$ denotes usage of a-priori knowledge about tempo[15]

Algorithm	A	$A*$
Lidy, Rauber and Pesenhofer	82.0	-
Gouyon and Dixon, 2004 [GD04]	67.6	-
Gouyon et al., 2004 [GDPW04]	78.9	90.1
Dixon et al., 2004 [DGW04]	85.7	96.0

training/test set split has been used on that database. Consequently, in the contest 210 test files had to be classified – based on the model trained on the 488 training instances – into the 8 available dance rhythms.

Our algorithm classified 82 % of the test files correctly and won the Rhythm Classification task. No other team participated in this ISMIR2004 task, nevertheless we can compare the evaluation result to other approaches which were published at the same time and tested on the same audio data (however, using 10-fold cross-validation instead of the 70:30 training/test set split). Table 5.4 shows this comparison.

The approach by Gouyon and Dixon [GD04] is based on tempo probability functions for each of the 8 ballroom dances and successive pairwise or three-class classification and reports 67.6 % overall Accuracy. Dixon et al. specifically address the problem of dance music classification and achieve a result of 96 % Accuracy when using a combination of various feature sets [DGW04]. Besides audio-based descriptors, the approach also incorporates a-priori knowledge about tempo and thus drastically reduces the number of possible classes for a given audio instance. The ground-truth-tempo approach has been previously described by Gouyon et al. [GDPW04], where classification based solely on the pre-annotated tempo attribute reached 82.3 % Accuracy (k-NN classifier, k=1). The paper also describes a variety of feature sets and reports 90.1 % Accuracy on the combination of MFCC-like descriptors with ground-truth tempo and 78.9 % Accuracy when using computed tempo instead.

[15] Result of Lidy et al. was achieved on the 70:30 training/test set split used in the contest, the other results using 10-fold cross validation.

5.5 Pre-MIREX 2005 Experiments and New Feature Sets

As a next step – prior to the ISMIR 2005 evaluation[16] – I did a profound evaluation of the processing steps involved in Rhythm Patterns feature extraction. Of particular interest were the psycho-acoustic transformations involved in the feature extraction and their influence to the results of classification tasks. From the results of these experiments, which are reported in [LR05], I identified both crucial and problematic transformations under the several psycho-acoustic processing steps. The experiments and the resulting conclusions are explained in detail in Section 5.5.2.

In 2005 I also developed two new feature sets: Statistical Spectrum Descriptors and Rhythm Histograms. In a further set of experiments I evaluated the performance of the two new feature sets on the same music databases and the same experiment setup as the Rhythm Patterns features and compared their performance to the one of the new Rhythm Patterns variants. The two feature sets have been described in Chapter 3, the evaluation experiments are described in Section 5.5.3.

The comparison of the three feature sets is elaborated in Section 5.5.5, which is followed by another experiment on combining the different feature sets and employing them together for classification in Section 5.5.6. As each feature set has different strengths and weaknesses, the combined approaches deliver improved results over the single feature sets.

Eventually, in Section 5.5.7 a comparison of the new feature sets and the combination approach is done with the results of the ISMIR 2004 Audio description contest as well as other published results on the same databases I used in my experiments. The music databases used will be presented in the following subsection, they are described in detail in Chapter 4. Finally, conclusions of all those experiments and evaluations are provided in Section 5.5.8.

5.5.1 Audio Collections and Experiment Setup

For the quantitative evaluation of the three feature sets I measured their performance in genre classification, similar as in the ISMIR 2004 Genre

[16] which was called MIREX from 2005 on

Classification contest (c.f. Section 5.4.2). The experiments were performed on three different audio collections in order to gain information about the generalization of the results to different music repositories and thus different genre taxonomies, or to possibly detect specific problems with certain music styles. Three reference audio collections have been used in order to be able to compare the results to other published measures: the GTZAN collection, the ISMIRrhythm collection and the ISMIRgenre collection. These collections are described in detail in Chapter 4.

The GTZAN collection consists of 1000 pieces of audio equi-distributed among 10 popular music genres (see Table 4.2). The ISMIRrhythm collection is the one used in the ISMIR 2004 Rhythm classification contest and consists of 698 30-second excerpts of 8 genres from ballroom dance music (see Table 4.3(b)). The ISMIRgenre collection is from the ISMIR 2004 Genre classification contest and contains 1458 songs from Magnatune.com, the pieces being unequally distributed over 6 genres (see Table 4.3(a)). The tables list the genres involved in each collection and also the numbers of songs in each genre category.

For classification, Support Vector Machines with pairwise classification were used, utilizing the Weka Machine Learning software. A 10-fold cross validation was performed in each experiment. From the experiments macro-averaged Precision (P^M) and Recall (R^M) are reported, as defined in Section 5.2. Macro-averaged Precision and Recall were used in order to make up for the unequal distribution of classes in the ISMIRgenre and ISMIRrhythm music collections. As globally comparable criterion the F_1 measure is reported, and for comparability to other studies Accuracy (A) is measured, which is the proportion of correctly classified songs to the total number of songs in a collection.

5.5.2 Evaluation of Psycho-Acoustic Transformations in Rhythm Patterns feature extraction

The Rhythm Patterns feature extraction algorithm includes a number of psycho-acoustically motivated transformations:

- conversion to the Bark scale (critical bands)
- a spreading function to account for spectral masking
- transformation into the Decibel scale

- computation of equal loudness (Phon scale)
- computation of specific loudness sensation (Sone scale)
- fluctuation strength weighting
- filtering and smoothing

Some of those transformations include logarithmic or non-linear transformations, some others even look-ups in tables, which makes them computationally complex. I conducted a set of experiments with the purpose to discover how important these psycho-acoustic transformations are to the performance of the feature set. The results from this experiments reveal information about which are crucial parts of the feature extraction as well as an indication of which transformations potentially pose problems to the performance of the feature set. Both of this is valuable information for further improvement of the feature extraction algorithm and has implications to audio feature extraction in general.

As in the new implementation of the Rhythm Patterns extractor – due to the introduction of many new parameters – several of the steps from the algorithm could be processed optionally, I conducted a reasonable number of experiments with different option settings regarding the transformations involved in the feature extraction process. Table 5.5 summarizes the steps involved in Rhythm Patterns calculation in each experiment. 'S' denotes steps for computing the Sonogram, 'R' denotes the steps on the modulation amplitudes representation computing the actual Rhythm Pattern, according to Figure 3.2. In order to be able to assess the generalization of the results on various genre taxonomies three different standard MIR audio collections have been used in the evaluation (see Section 5.5.1).

Table 5.6 provides an overview of the experiments. Each experiment is identified by a letter. The table lists the steps of the feature extraction process involved in each experiment. Experiment A represents the baseline, where all the feature extraction steps are involved. Experiments K through N completely omit the transformations into the dB, Phon and Sone scales. Experiments G to I and K to Q extract features from the audio without accounting for Spectral Masking effects. A number of experiments evaluates the effect of filtering/smoothing and/or the fluctuation strength weighting.

In Table 5.7 the results from experiments A through Q on the three audio collections are presented (best and second-best result in each column

Table 5.5: Summarization of the steps for computation of Rhythm Pattern features

step	description
S1	Power Spectrum (FFT)
S2	Critical bands (Bark)
S3	Spectral Masking
S4	Loudness (dB)
S5	Equal Loudness (Phon)
S6	Specific Loudness Sensation (Sone)
R1	Modulation Amplitudes (FFT)
R2	Fluctuation Strength
R3	Filtering and Smoothing

Table 5.6: Experiment IDs and the steps of the Rhythm Patterns feature extraction process involved in each experiment

step	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
S2	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
S3	×	×	×	×	×	×				×							
S4	×	×	×	×	×	×	×	×	×						×	×	×
S5	×	×	×	×	×		×	×							×	×	×
S6	×	×	×	×			×								×	×	×
R1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
R2	×	×			×	×	×	×	×	×	×	×			×		
R3	×		×		×	×	×	×	×	×	×		×			×	

are printed in boldface). From the results of the experiments several interesting observations can be made. Probably the most salient observation is the low performance of the experiments J through N (with the exception of Precision on the ISMIRgenre collection). These experiments do not involve transformation into decibel scale nor successive transformation into the Phon and Sone scales. Also, experiments E and F as well as H and I deliver quite poor results, at least on the GTZAN and ISMIRgenre data sets. Those experiments perform decibel transformation but skip the transformation into Phon and/or Sone. All these results indicate clearly that transformation into the logarithmic decibel scale is very important, if not essential, for the audio feature extraction and subsequent classification or retrieval tasks. The successive application of the equal loudness curves (i.e.

Table 5.7: Results of the Rhythm Patterns feature extraction experiments on evaluation of psycho-acoustic transformations: Macro-averaged Precision (P^M), macro-averaged Recall (R^M), F_1 measure and Accuracy (A) in % (highest and second highest value in each column are boldfaced)

	GTZAN				ISMIRrhythm				ISMIRgenre			
Exp.	P^M	R^M	F_1	A	P^M	R^M	F_1	A	P^M	R^M	F_1	A
A	58.5	58.5	58.5	58.5	82.5	81.3	81.9	81.7	59.8	56.1	57.9	71.0
B	62.6	62.3	62.5	62.3	**83.4**	81.6	82.4	**82.4**	62.4	61.8	62.1	72.6
C	59.7	59.4	59.5	59.4	**83.4**	**82.3**	**82.8**	**82.8**	59.7	56.3	57.9	71.2
D	62.6	62.3	62.5	62.3	**83.2**	81.4	82.3	82.2	62.5	61.6	62.0	72.6
E	55.5	55.8	55.7	55.8	81.7	80.9	81.3	81.4	59.6	58.0	58.8	70.4
F	53.6	53.6	53.6	53.6	82.0	81.1	81.6	81.7	57.2	54.7	55.9	68.2
G	**63.0**	**62.9**	**62.9**	**62.9**	82.6	81.6	82.1	82.0	65.6	60.8	63.1	73.7
H	59.1	59.5	59.3	59.5	81.9	80.6	81.3	81.4	59.6	58.6	59.1	71.5
I	59.7	60.2	60.0	60.2	82.4	81.0	81.7	81.8	59.4	57.9	58.6	70.3
J	53.1	52.3	52.7	52.3	74.1	72.7	73.4	73.5	64.5	52.0	57.6	69.3
K	53.9	53.1	53.5	53.1	74.1	72.3	73.1	73.4	**66.8**	52.7	58.9	70.0
L	55.1	54.4	54.7	54.4	67.1	66.5	66.8	67.8	63.8	54.5	58.8	69.6
M	54.5	53.9	54.2	53.9	74.9	72.4	73.6	73.5	**66.4**	52.2	58.5	69.2
N	55.4	54.7	55.0	54.7	67.0	66.3	66.6	67.3	63.4	53.8	58.2	69.1
O	**64.2**	**64.4**	**64.3**	**64.4**	80.7	79.3	80.0	80.1	65.1	**64.5**	**64.8**	**75.0**
P	60.5	60.5	60.5	60.5	**83.2**	**81.9**	**82.5**	82.2	66.2	61.6	63.8	73.9
Q	**64.2**	**64.4**	**64.3**	**64.4**	81.6	80.2	80.9	81.0	64.9	**64.1**	**64.5**	**74.9**

Phon transformation) and the calculation of Sone values appear also as important steps during feature extraction (experiment A compared to E and F, or experiment G compared to H and I).

Spectral Masking (step S3) was the subject of numerous experiments. I wanted to measure the influence of the use or omission of the spreading function for Spectral Masking in combination with variations in the other feature extraction steps. Table 5.7 clearly shows, that most experiments without Spectral Masking achieved better results. The ISMIRrhythm collection constitutes an exception to this. Nevertheless, the degradation of results incorporating spectral masking raises the question whether the Spectral Masking spreading function is inappropriate for music of certain styles. However, it is also possible, that the application of the Spectral Masking spreading function is problematic when using compressed music (e.g. in MP3 format)[17], as many encoders already consider Spectral Masking when

[17] The ISMIRrhythm data set was available in RealAudio format, the ISMIRgenre data set in MP3 format, both were decoded to WAV format. The GTZAN data set was available in uncompressed AU format.

encoding the music.

Further focuses of investigation were the effects of the fluctuation strength weighting curve (step R2) and the filtering/smoothing of the Rhythm Patterns (step R3). Both the GTZAN and ISMIRgenre collections are classified considerably better without the use of gradient filter and smoothing. The ISMIRrhythm collection, however, shows contrary results. Its results improve when omitting the fluctuation strength weighting, but degrade when filtering and smoothing is omitted.

One can observe in several experiments that the ISMIRrhythm collection behaves quite contrary to the two other collections. However, the results on the ISMIRrhythm collection were already considerably better than those on the two other collections from the beginning using the baseline algorithm. The reason why this collection behaves differently might be that the results are already at a high level and variations in the algorithm only cause small fluctuations on the results. On the other hand, contrary to the GTZAN collection and ISMIRgenre collection, ISMIRrhythm contains music from 8 different dance music styles. The discrimination of ballroom dances relies heavily, if not exclusively, on rhythmic structure, which makes the Rhythm Patterns feature set an ideal descriptor (and thus justifies the good results). Apparently, smoothing the Rhythm Patterns is important for making dances from the same class with slightly different rhythms more similar – whereas in the two other collections, filtering and smoothing has negative effects. The ISMIRrhythm set appears to be independent of the Spectral Masking effects. Best results with ISMIRrhythm were retrieved with experiment C, which omits fluctuation strength weighting (step R2), closely followed by experiment P, which additionally omits spectral masking (step S3).

For the GTZAN and ISMIRgenre collections best results both in terms of F_1 measure and Accuracy were achieved in experiment O, which is the original Rhythm Patterns feature extraction without Spectral Masking (S3) and without filtering and smoothing (R3).

5.5.3 Evaluation of Statistical Spectrum Descriptors

Statistical Spectrum Descriptors (SSD) are derived from a psycho-acoustically transformed Bark-scale spectrogram and comprise several statistical moments, which are intended to describe fluctuations on a number of critical frequency bands. The transformations involved in SSD computation

Table 5.8: Results of the experiments with Statistical Spectrum Descriptors (best results bold).

	GTZAN				ISMIRrhythm				ISMIRgenre			
Exp.	P^M	R^M	F_1	A	P^M	R^M	F_1	A	P^M	R^M	F_1	A
SSD (S2) / mean	60.9	60.2	60.5	60.2	36.6	21.1	26.7	25.6	40.6	25.6	31.4	51.6
SSD (S2) / median	57.7	57.0	57.3	57.0	43.5	40.0	41.7	43.8	68.2	49.9	57.6	67.8
SSD (S6) / mean	**72.9**	**72.7**	**72.8**	**72.7**	**54.4**	52.8	53.6	54.7	**76.9**	**68.0**	**72.2**	**78.5**
SSD (S6) / median	71.6	71.3	71.4	71.3	**54.4**	**53.8**	**54.1**	**55.4**	75.8	66.7	71.0	77.5

are outlined in Figure 3.2 and are explained in Section 3.2.6.

In the experiments with the Statistical Spectrum Descriptor (SSD) I mainly investigated the performance of the features depending on which position in the Rhythm Patterns feature extraction process they are computed. Two positions were chosen to test the SSD: First, the statistical measures are derived directly after step S2, when the frequency bands of the audio spectrogram have been grouped to critical bands. In the second experiment, the features are calculated after the critical bands spectrum had undergone logarithmic dB transformation as well as transformation into Phon and Sone, i.e. after step S6. In order to find an adequate representation of an audio track through a Statistical Spectrum Descriptor, I evaluated both the calculation of the mean and the median of all segments of a track.

Table 5.8 gives the results of the four experiment variants. The results clearly indicate, that in any case the calculation after step S6 is superior to deriving the SSD already at the earlier stage, step S2. Consequently, as in the experiments with the Rhythm Patterns feature set, logarithmic transformation appears to be essential for the results of the content-based audio descriptors. Comparing the summarization of an audio track by mean and by median, results of the GTZAN and ISMIRgenre collection argue for the use of the mean. Again, the ISMIRrhythm collection indicates contrary results, however the differences in result measures vary only between 0.04 and 1.4 percentage points.

Note, that the SSD feature set calculated after step S6 outperforms the Rhythm Patterns descriptor both on the GTZAN and ISMIRgenre collections. This is especially remarkable as the Statistical Spectrum Descriptors, with 168 feature dimensions, have a dimensionality 8.5 times lower than the Rhythm Patterns feature set.

5.5.4 Evaluation of Rhythm Histogram Features

Rhythm Histogram features (RH) describe global rhythmic content of a piece of audio by a measure of energy per modulation frequency (c.f. Section 3.2.7). They are calculated from the time-invariant representation of the Rhythm Patterns. In the next set of experiments I tried to evaluate different performance when computing the Rhythm Histogram Features after feature extraction step R1, R2 or R3, respectively. Evaluation showed, that regardless to the stage, RH features virtually always produce equal results. Therefore I omit a separate table with detailed results; performance of the Rhythm Histogram features can be seen in the row denoted 'RH (R1)' of Table 5.9.

RH features tested on the ISMIRrhythm collection achieve nearly the results of the Rhythm Patterns feature set. Note that the dimensionality (60) is 24 times lower than that of RP features. Performance on GTZAN and ISMIRgenre collections is rather low, nevertheless the Rhythm Histogram feature set seems eligible for audio content description and shows its strengths in combination with SSD features (c.f. Section 5.5.6).

5.5.5 Comparison of Feature Sets

Table 5.9 displays a comparison of the baseline Rhythm Patterns (RP) algorithm (experiment A) to the best results of the Rhythm Patterns extraction variants, the Statistical Spectrum Descriptor (SSD) and the Rhythm Histogram features (RH). Best results in Rhythm Patterns extraction were achieved with the GTZAN, ISMIRrhythm and ISMIRgenre audio collections in experiments O, C, and O respectively (c.f. Section 5.5.2). Accuracy was 64.4 %, 82.8 %, and 75.0 %, respectively. The Statistical Spectrum Descriptor performed best when calculated after psycho-acoustic transformations, and taking the simple mean of the segments of a piece of audio. Accuracy was 72.7 %, 54.7 %, and 78.5 % on the GTZAN, ISMIRrhythm and ISMIRgenre data sets, respectively, which exceeds the Rhythm Patterns feature set in 2 of the 3 collections. Rhythm Histogram Features achieved 44.1 %, 79.9 %, and 63.2 % Accuracy, which rival the Rhythm Patterns features regarding the ISMIRrhythm data collection. The conclusions of these series of experiments was that a combination of these feature sets would potentially enhance classification results.

5.5.6 Combination of Feature Sets

As the the three different feature sets evaluated in the previous experiments performed in divergent manner on different data sets, a combination of feature sets was the next logical step. The assumption was that each feature set has different strengths and weaknesses and that a combined approach would deliver improved results over the single feature sets. I did a number of experiments on all possible 2-set combinations as well as the combination of all three feature sets, with the goal to further improve performance in classification tasks. Results of the experiments are included in Table 5.9.

The combination of Rhythm Patterns features with the Statistical Spectrum Descriptor achieves 72.3 % Accuracy on the GTZAN data set, which is slightly lower than the performance of the SSD alone. Contrary, on the ISMIRrhythm data set, the combination achieves a slight improvement. On the ISMIRgenre audio collection, this combination results in a significant improvement and achieves the best result of all experiments on this data set (80.3 % Accuracy). Combination of Rhythm Patterns features with Rhythm Histogram Features changes the results of the Rhythm Patterns features only insignificantly, a noticeable improvement can be seen only in the ISMIRrhythm data set, which is the data set where the Rhythm Histogram features performed best.

Very interesting are the results of combining the Statistical Spectrum Descriptor with Rhythm Histogram features: With the GTZAN collection, this combination achieves the best Accuracy (74.9 %) of all experiments (including the 3-set experiment). The result on the ISMIRrhythm collection is comparable to the best Rhythm Patterns result. The 2-set combination without Rhythm Patterns features performs also very well on the ISMIRgenre data set, achieving the best F_1 measure (73.3 %). There is a notably high Precision value of 76.7 %, however, Recall is only at 70.2 %. Accuracy is 79.6 % and thus slightly lower than in the Rhythm Patterns + SSD combination.

Finally, I investigated the combination of all three feature sets, which further improved the results only on the ISMIRrhythm data set. Accuracy increased to 84.2 %, compared to 82.8 % using only the Rhythm Patterns features. As stated before, results on the ISMIRrhythm collection were rather high from the beginning, consequently improvements on classification in this data set were moderate.

Table 5.9: Comparison of feature sets and combinations (best results boldfaced)

	GTZAN				ISMIRrhythm				ISMIRgenre			
feature set(s)	P^M	R^M	F_1	A	P^M	R^M	F_1	A	P^M	R^M	F_1	A
RP(A)	58.5	58.5	58.5	58.5	82.5	81.3	81.9	81.7	59.8	56.1	57.9	71.0
RP(best) (O/C/O)	64.2	64.4	64.3	64.4	83.4	82.3	82.8	82.8	65.1	64.5	64.8	75.0
SSD (S6) (mean)	72.9	72.7	72.8	72.7	54.4	52.8	53.6	54.7	**76.9**	68.0	72.2	78.5
RH (R1)	43.6	44.1	43.8	44.1	82.1	79.1	80.6	79.9	41.6	39.2	40.4	63.2
RP(best)+SSD	72.2	72.3	72.2	72.3	84.4	82.9	83.6	83.5	72.3	**72.0**	72.2	**80.3**
RP(best)+RH	64.1	64.2	64.1	64.2	84.5	83.1	83.8	83.7	65.3	64.6	64.9	75.5
SSD+RH	**74.8**	**74.9**	**74.8**	**74.9**	83.1	81.4	82.3	82.7	76.7	70.2	**73.3**	79.6
RP(best)+SSD+RH	72.3	72.4	72.3	72.4	**85.0**	**83.4**	**84.2**	**84.2**	71.9	71.3	71.6	80.0

5.5.7 Comparison with Other Results

GTZAN data set

The GTZAN audio collection was assembled by George Tzanetakis and used in experiments in his dissertation [Tza02]. The original collection was organized in a three level hierarchy intended for discrimination into speech/music, classification of music into 10 genres and subsequent classification of the two genres classical and jazz into subgenres. In our experiments we used the organization of 10 musical genres at the second level, and thus compare our results to the performance reported in [Tza02] on that level. The best classification result reported was 61 % Accuracy (4 % standard deviation on 100 iterations of a 10-fold cross validation) using the 30 dimensional MARSYAS genre features and Gaussian Mixture Models.

Li et al. [LOL03] used the same audio collection in their study and compare “Daubechies Wavelet Coefficient Histograms” (DWCHs) to combinations of MARSYAS features. DWCHs achieved 74.9 % classification Accuracy in a 10-fold cross validation using Support Vector Machines (SVM) with pairwise classification and 78.5 % Accuracy using SVM with one-versus-the-rest classification.

The best performance I achieved in my experiments (in which I uniformly used pairwise classification) was 74.9 % – an improvement of 16.4 percentage points regarding the baseline Rhythm Patterns features, yet with much lower-dimensional features. This is equal to the result which Li et al. achieved with SVM using pairwise classification.

Table 5.10: Comparison of SSD+RH result with other results on the GTZAN audio collection: Accuracy in %

	A
Tzanetakis, 2002 [Tza02] (GMM)	61.0
Li et al., 2003 [LOL03] (SVM, pairwise classification)	74.9
Li et al., 2003 [LOL03] (SVM, one-vs-the-rest classification)	78.5
SSD+RH (in this thesis) (SVM, pairwise classification)	74.9

ISMIRrhythm data set

Already in Section 5.4.4 I compared my result from the ISMIR 2004 Rhythm Classification contest to other published results on the same data collection. Some of the approaches use a-priori tempo knowledge about the dance rhythms and I therefore listed these results in a separate column of Table 5.4. In my current evaluation experiments, the combination of the three feature sets (RP, SSD and RH) achieved the best performance on the ISMIRrhythm data set, with 84.2 % Accuracy. This is an improvement of 2.5 percentage points over the baseline algorithm, which was at an already very high level on this data set. Dixon et al. [DGW04] report 85.7 % Accuracy when not using ground-truth tempo information, thus I came very close to their result. Furthermore the results of my approach are higher than two of the other approaches investigated in Section 5.4.4, evident from Table 5.4.

ISMIRgenre data set

The ISMIRgenre data set was assembled for the ISMIR 2004 Genre Classification contest (see Section 5.4.2). Results from the Genre Classification contest are shown in Table 5.1 in terms of Accuracy and Accuracy normalized by the genre frequency, which is the same as macro-averaged Recall (R^M) given in Table 5.9. However, in order to be able to compare the current evaluation experiment result to the values from the 2004 contest, instead of a 10-fold cross-validation I had to repeat the experiment on the combination of RP(O)+SSD features (which delivered the best result on the ISMIRgenre data set) using the same training and test set partitioning as in the 2004 Genre Classification contest (50:50). The result was 79.7 % Accuracy and 70.4 % normalized Accuracy (slightly lower values as using

10-fold cross-validation, c.f. Table 5.9). Compared to the results of the 2004 Genre Classification contest this approach would have surpassed two other approaches, making it theoretically rank second place (c.f. Table 5.1). Though not surpassing the winner of the 2004 contest, the results of this evaluation represent a substantial improvement to the approach submitted to the 2004 Audio Description contest, with an improvement of 9.3 percentage points.

5.5.8 Conclusions

In this Section I performed a study evaluating three feature sets and combinations of them, evaluated on three standard benchmark music collections, and compared the results to published performance measures of other researchers on the same data sets. These feature sets are intended for different application scenarios in music similarity retrieval, one of them is automatic music classification. Performance on all experiments in this Section was measured by the results of a music genre classification task.

First, I performed a series of experiments on the importance of psycho-acoustic transformations in within the computation of Rhythm Patterns audio features. Experiments confirmed that the inclusion of a number psycho-acoustic transformations results in a substantial improvement of classification accuracy. In particular, these are the findings of the experiments on psycho-acoustic transformation steps:

- The implementation of Spectral Masking in the feature extraction might pose a potential issue in audio description.
- Transformation into the logarithmic decibel scale is crucial.
- Implementation of equal loudness curves, which transforms the spectrum into the Phon scale is very important.
- Computation of specific loudness sensation in terms of the Sone scale is very important.
- The weighting of fluctuation strength according to a psycho-acoustic model has been identified to have quite unpredictable influence depending on the music collection used.

- Applying filtering and smoothing is beneficial to music databases in which beat is the most important factor for distinguishing genres, but may have negative effects in other music collections.

In an additional set of experiments I evaluated two newly developed feature sets, namely Statistical Spectrum Descriptors and Rhythm Histograms, and compared their performance on music genre classification to the optimal setting of the Rhythm Patterns feature set according to the previous experiments. Both Statistical Spectrum Descriptors and Rhythm Histograms have specific strengths, which made them outperform the Rhythm Patterns features on two respectively one of the three test databases. A combination of the feature sets was the logical consequence and the subject of further experiments, in which I investigated different combination settings and whether they would achieve additional improvements in the classification task. In terms of Accuracy, the combination of (improved) RP and SSD features performed best on one collection, the (much lower dimensional) combination of SSD and RH features on another collection and the combination of all three sets on the third collection.

Overall improvement, regarding best Accuracy values achieved in each data collection compared to baseline Rhythm Patterns algorithm (experiment A in Table 5.7), was +16.4 percentage points on the GTZAN music collection, +2.5 percentage points on the ISMIRrhythm collection and +9.3 percentage points on the ISMIRgenre music collection.

The influence of the segment sizes, FFT window sizes, the Bark scale and alternative frequency groupings has been evaluated in another set of experiments by Laister [Lai06]. The filtering and weighting processes should be investigated in an additional set of experiments using other audio databases.

5.6 MIREX 2005

The second evaluation campaign in MIR research was held in parallel to the ISMIR 2005 conference and was now called MIREX – Music Information Retrieval Evaluation eXchange. It was organized by IMIRSEL (International Music Information Retrieval Systems Evaluation Laboratory), the project that was started in 2002 to unite the efforts of establishing a common MIR evaluation forum. The MIREX evaluation is open to any group or individual who wants to participate with their algorithm(s) and system(s) in an evaluation and comparison of state-of-the-art MIR algorithms. Moreover, any participant can propose new tasks to the MIREX forum. In 2005 the following list of tasks was available where participants could submit their algorithms to:[18]

- Audio Artist Identification
- Audio Drum Detection
- Audio Genre Classification
- Audio Melody Extraction
- Audio Onset Detection
- Audio Tempo Extraction
- Audio and Symbolic Key Finding
- Symbolic Genre Classification
- Symbolic Melodic Similarity

The number of tasks had increased from 5 to 10 over the ISMIR 2004 Audio Description Contest. The number of problems addressed had grown and both the symbolic and audio-based Music Information Retrieval domains were represented.

My main interest was the evaluation of the new approaches based on the experiments described in the previous section on the Audio Genre Classification task. As in the 2004 evaluation we submitted the same approach also to the Audio Artist Identification task in order to see the applicability and generalization to a greater number of classes. However, runtime was limited to 24 hours and our submission to the Artist Identification task ran out of time due to a scaling problem.

5.6.1 Submitted Algorithm

From the Pre-MIREX 2005 experiments a number of observations have been made which were important for the participation in the MIREX 2005 evaluation. For instance, it was substantial to include the psycho-acoustic transformations for Sonogram computation (c.f. Sections 3.2.5 and 5.5.2). It was also obvious from the pre-evaluation that we would submit a combination of

[18] `http://www.music-ir.org/mirex2005/index.php/Main_Page`

feature sets. However, different combination settings performed differently depending on what music database was used.

MIREX 2005 allowed each participant to submit multiple systems. In accordance with the MIREX 2005 guidelines, we therefore submitted three different combinations of feature sets. This allowed us to participate with three different approaches in a state-of-the-art comparison and at the same time to evaluate the approaches individually on two different MIREX 2005 databases used in Audio Genre Classification.

I was interested particularly in the performance of the new feature sets which had shown contrary results: SSD and Rhythm Histograms. The combination of SSD and Rhythm Histograms had by far the lowest dimensionality, which results in much lower computation time in the classification task. This combination was also expected to represent a more generalized feature set with potentially better results in a broader variety of musical styles. Moreover, the questions was, whether classification without the much higher-dimensional Rhythm Patterns feature set could achieve comparable results. Besides, we submitted a combination of RP and SSD features as well as the 3-set combination. The following combinations of feature sets have been submitted to MIREX 2005:

- Rhythm Patterns + SSD (1608 dimensions)
- SSD + Rhythm Histograms (228 dimensions)
- Rhythm Patterns + SSD + Rhythm Histograms (1668 dimensions)

The output of the feature extractors was combined by concatenating the attributes of the individual feature sets into a single combined feature vector.

For learning and classification we used linear Support Vector Machines in the SMO implementation of the WEKA Machine Learning Software with pairwise classification, as in the previous evaluations. As far as scaling of the submitted approaches is concerned: The feature extraction part scales linearly with the number of audio instances and the classification part scales quadratically with the dimensions of the feature vectors.

5.6.2 Audio Genre Classification

In the MIREX 2005 Audio Genre Classification task[19] 15 algorithms from 12 participating teams or individuals have been evaluated on two different music databases:

- Magnatune[20]: 10 genres, 1005 training files, 510 testing files
- USPOP 2002: 6 genres, 940 training files, 474 testing files

The list of genres and number of tracks per genre in the test sets of both collections are given in Table 4.4. The collections are described in Sections 4.2.4 and 4.2.5. The audio files were available with a sampling frequency of 44,100 or 22,050 Hz, mono or stereo, as desired by each participant.

Performances of the participating systems have been evaluated separately on these data sets, however, an overall score has been calculated from both for the official end ranking of MIREX 2005.

Multiple evaluation measures were computed on both audio databases: raw classification Accuracy and Accuracy normalized by the number of tracks per genre. While the USPOP dataset was categorized by a single genre level, the Magnatune dataset was organized by a hierarchical genre taxonomy. The hierarchical genre taxonomy combined the pairs of Jazz & Blues, Rock & Punk, Folk & Ethnic as well as 3 'electronical' genres to the same super-genre, while classical music constituted a genre of its own. The genre hierarchy is depicted in Figure 4.1.

In the evaluation of the Magnatune database, additional measures on hierarchical classification were computed: in this case, less penalty was given to mis-classification into a genre which belonged to the correct super-genre. In principal, for a genre hierarchy of n levels a score of $1/n$ is given for each correctly identified level. For instance, considering the MIREX 2005 Magnatune hierarchy depicted in Figure 4.1, a mis-classification of a Rock piece as Punk would give the score of 1/2 point, because the super-genre of 'Rock & Punk' was matched, while a mis-classification of Rock as Blues would give 0 points.

The overall measure for MIREX 2005 Audio Genre Classification was calculated by the mean of the Magnatune hierarchical classification Accu-

[19] http://www.music-ir.org/mirex2005/index.php/Audio_Genre_Classification

[20] http://www.magnatune.com, the same source as in the ISMIR 2004 Genre Classification task has been used, but a different set of music files.

Table 5.11: MIREX 2005 Audio Genre Classification overall results (mean Accuracy in % from 2 benchmark data sets).

Rank	Participant	Result
1	Bergstra, Casagrande & Eck (2)	82.34
2	Bergstra, Casagrande & Eck (1)	81.77
3	Mandel & Ellis	78.81
4	West, K.	75.29
5	Lidy & Rauber (SSD+RH)	75.27
6	Pampalk, E.	75.14
7	Lidy & Rauber (RP+SSD)	74.78
8	Lidy & Rauber (RP+SSD+RH)	74.58
9	Scaringella, N.	73.11
10	Ahrendt, P.	71.55
11	Burred, J.	62.63
12	Soares, V.	60.98
13	Tzanetakis, G.	60.72

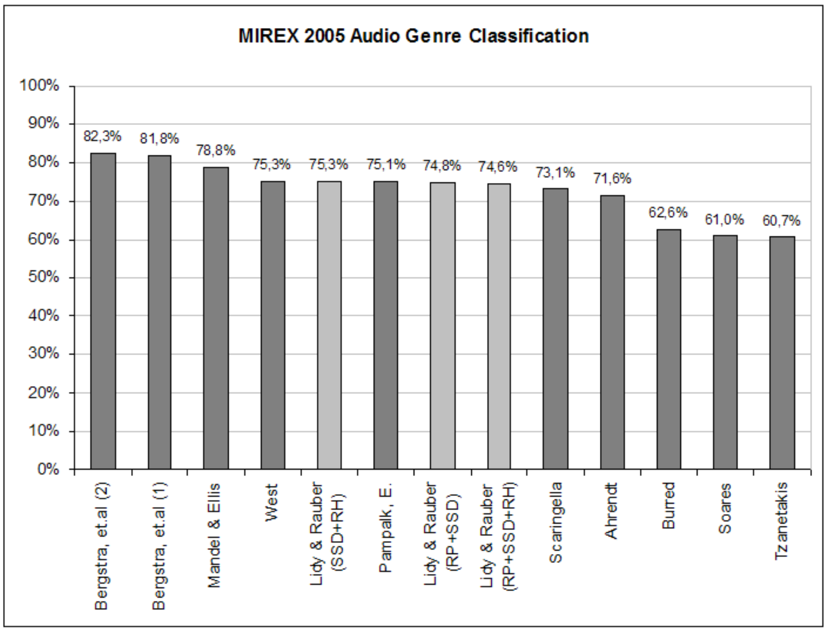


Figure 5.1: Diagram of MIREX 2005 Audio Genre Classification overall results

Table 5.12: Magnatune data set: ranking and hierarchical classification Accuracy

Rank	Algorithm	A^h
5	Lidy & Rauber (RP+SSD)	71.08
6	Lidy & Rauber (RP+SSD+RH)	70.88
7	Lidy & Rauber (SSD+RH)	70.78

Table 5.13: USPOP data set: ranking and raw classification Accuracy

Rank	Algorithm	A
5	Lidy & Rauber (SSD+RH)	79.75
7	Lidy & Rauber (RP+SSD)	78.48
9	Lidy & Rauber (RP+SSD+RH)	78.27

racy and the USPOP raw classification Accuracy. Our best result achieved 75.27 %, which was the 5th rank. The overall rankings and results of our three algorithms are given in Table 5.11 and shown in Figure 5.1.

From our three submitted systems, the feature combination with the lowest dimensionality (SSD + RH) achieved the best results of our approaches, however, all three of our variants achieved very similar results (c.f. Figure 5.1). Also submissions of other participants (Mandel & Ellis, West, Scaringella, Pampalk, Ahrendt) achieved very similar results (at least on one of the data sets), and the question for significant differences opened the call for additional statistical significance tests – which were then introduced in MIREX 2006. Only the algorithms by Bergstra, Casagrande & Eck (ranked 1st and 2nd, with 82.34 % and 81.77 % overall Accuracy, respectively), as well as Mandel & Ellis with 85.65 % raw Accuracy on the USPOP data set, seem to be significantly ahead.

Bergstra, Casagrande & Eck extracted a relatively large number of timbre features (MFCCs, RCEPS, ZCR, LPC, Rolloff, among others) at an intermediate time scale (every 13.9 seconds) and calculated mean and variance of the features for each segment. They used AdaBoost.MH [FS95] for classification, in variant (1) they boosted decision stumps and in variant (2), which was ranked 1st in the MIREX 2005 evaluation, they boosted 2-level trees. They classified the features extracted from each time segment independently and determined the class label for a song by averaging the outputs of the meta-feature classifiers.

Regarding our three variants, the ranking order varies depending on the music database – c.f. Tables 5.12 and 5.13. However, considering the very low difference in Accuracy, as a conclusion it might be better to choose the SSD+RH combination for Genre Classification in the future, due to performance reasons.

The comprehensive evaluation, details about the submitted approaches as well as confusion matrices of each individual result can be obtained from the MIREX 2005 Audio Genre Classification results web page[21].

Investigation of the confusion matrices enables to identify problematic genres and might potentially give hints for future improvements. In the confusion matrix of the USPOP database (6 genres, see Table 5.14) the genre with lowest Accuracy was Reggae, often confused with Rap & Hip-hop or Electronica & Dance. Differences between the three algorithm variants show, that potential improvement in discrimination by using other features is possible. The SSD+RH feature combination for instance recognized Electronic and Rap music better than the other combinations, while Reggae music is discriminated better when RP features are included, as can be seen from the confusion matrices of the other feature sets which are available on the results web site. The low Accuracy on Reggae might also be a result of the low number of Reggae instances in the database (54 in total, 36 for training and 18 for testing). Contrarily, the genre New Age has been classified with 90.5 to 95.2 % Accuracy, although there are only 61 pieces in total in the database. Electronica & Dance as well as Country pieces were often mis-classified as Rock pieces.

The best discriminated classes within the Magnatune data set (10 genres, see Table 5.15) were Blues and Classical with over 97 % Accuracy. The SSD+RH approach also achieved 97 % on the Punk genre. The genre with the worst performance was New Age, which was more often classified as Ethnic music. Note that in the USPOP database New Age was the *best* recognized genre. The SSD+RH approach also heavily confused Jazz music with Ethnic music. The reason might be the sometimes very blurry genre boundaries, especially with genres like Ethnic or New Age. However, as with the USPOP database, Electronic music has been confused with Rock music,

[21] `http://www.music-ir.org/evaluation/mirex-results/audio-genre/index.html`

Table 5.14: Confusion Matrix of SSD+RH algorithm on USPOP data set (%). Abbreviations are the first two letters of the genres listed in Table 4.4(b). Columns: true classes, rows: predictions

	Co	El	Ne	Ra	Re	Ro
Co	76.2	1.5	0.0	0.9	0.0	6.6
El	0.0	64.2	0.0	4.3	16.7	4.8
Ne	0.0	3.0	90.5	0.0	0.0	0.0
Ra	0.0	9.0	0.0	88.0	22.2	4.8
Re	0.0	0.0	0.0	2.6	50.0	0.0
Ro	23.8	22.4	9.5	4.3	11.1	83.8

Table 5.15: Confusion Matrix of SSD+RH algorithm on Magnatune data set (%). Abbreviations are the first two letters of the genres listed in Table 4.4(a). Columns: true classes, rows: predictions

	Am	Bl	Cl	El	Et	Fo	Ja	Ne	Pu	Ro
Am	70.6	0.0	0.0	2.4	6.0	0.0	4.6	11.8	0.0	2.4
Bl	0.0	97.1	0.0	0.0	0.0	8.3	9.1	5.9	0.0	3.6
Cl	5.9	0.0	97.5	0.0	10.8	0.0	0.0	2.9	0.0	2.4
El	8.8	0.0	0.0	62.2	7.2	12.5	9.1	14.7	0.0	13.1
Et	8.8	0.0	2.5	9.8	55.4	8.3	22.7	32.4	2.9	8.3
Fo	0.0	0.0	0.0	2.4	9.6	58.3	0.0	0.0	0.0	2.4
Ja	0.0	0.0	0.0	1.2	0.0	0.0	31.8	0.0	0.0	0.0
Ne	5.9	0.0	0.0	0.0	4.8	0.0	4.6	23.5	0.0	1.2
Pu	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.1	4.8
Ro	0.0	2.9	0.0	22.0	6.0	12.5	18.2	8.8	0.0	61.9

which calls for further investigation of the feature representations.

A big advantage of joint evaluation campaigns is that detailed results can be compared directly. From the confusion matrices on the results web pagewe see, that also many other participants had problems with the confusion of Electronic music with Rock music and/or New Age with Ethnic music.

5.7 Pre-MIREX 2006 Distance Metric Evaluation

5.7.1 New Task Definitions for MIREX 2006

Participants in the MIREX 2005 classification tasks had put many effort to optimize classifier parameters and to tweak classification algorithms, e.g. kernels of Support Vector Machines, to improve the results on Genre Classification and/or Artist Identification. This, however, was not the intention of the Music Information Retrieval Evaluation eXchange, which aims at being an evaluation campaign for Audio and Symbolic Music Description rather than for Machine Learning algorithms.

As a consequence, in the Pre-MIREX 2006 phase it was discussed to evaluate "Music Similarity" in a different manner than a Genre Classification task. The idea was to evaluate algorithms in a retrieval task, in which algorithms would have to return a given number of songs "similar" to a query song from a music database. The issue of each of the participants using different classification approaches would then be obsolete, and the evaluation based on the percentage of top k retrieved songs having the same genre label as the query song would essentially come close to a classification by k-nearest neighbors. Furthermore, one goal for MIREX 2006 was to massively increase the size of the music database(s) in order to come closer to evaluations of real-world scenarios, which would make complex Machine Learning algorithms very costly. It was defined as a requirement for the MIREX 2006 tasks to compute an $N \times N$ distance matrix between all N songs in a database in order to enable numerous evaluation statistics for MIREX 2006 as well as for post-MIREX evaluations.

Moreover, participants in the Pre-MIREX 2006 Music Similarity mailing list[22] discussed the realization of a large-scale human listening test for MIREX 2006 Audio Music Similarity and Retrieval. The top k retrieval results of a number of query songs would be judged by human graders giving scores of "similarity" to each retrieved song. The actual problem definition or application of the "Music Similarity" task, however, was left open, and so was the interpretation of the term "similarity". Music similarity can be useful in a number of different tasks, such as playlist generation, retrieval of "similar" sounds from a database, clustering, classification, etc. The actual dimension(s) of "similarity" which should be applied in the MIREX 2006

[22] https://mail.lis.uiuc.edu/mailman/listinfo/mrx-com00

Music Similarity task was left open to both the participants and the human evaluators, which were free to decide if "similarity" would include similar rhythm, melody, tempo, etc. or not.

5.7.2 Evaluation of Distance Metrics for Music Similarity Retrieval

Until now I had employed the feature sets under investigation either to classification tasks or in clustering-based applications (see Chapter 6), but not directly in (ranked) retrieval. For clustering using Self-Organizing Maps (c.f. Section 6.2) the Euclidean distance was used for neighborhood calculation. A study of different approaches for distance computations had yet to be done.

A new Java implementation of the RP, RH and SSD feature sets was submitted to MIREX 2006 (see Section 5.8.1) and I also used features computed with the Java version in the following study of different distance metrics. The feature space of these descriptors is high-dimensional and it is not obvious which method to use for distance computation. More advanced approaches analyze distribution and evolution of features computed at several temporal positions in the music. Among them are Gaussian Mixture Models or Hidden Markov Models. However, I did not use these more complex approaches and instead studied different metrics which could be applied directly within the feature space.

A common set of distance metrics for normed vector spaces are the Minkowski metrics, derived from the L_p norms, and computed as:

$$d_p(x,y) = L_p(x,y) = (\sum_{i=1}^{d} |x_i - y_i|^p)^{1/p} \tag{5.7}$$

For $p \geq 1$ the Minkowski metrics have the following properties:

1. positivity: $d_p(x,y) \geq 0$
2. symmetry: $d_p(x,y) = d_p(y,x)$
3. non-degeneracy: $d_p(x,y) = 0 \Leftrightarrow x = y$
4. they fulfill the triangle inequality: $d_p(x,y) \leq d_p(z,x) + d_p(z,y)$

In theory, p can be any real value, however, for $p < 1$ the triangle inequality does not hold any more.

A value of $p = 1$ results in the so-called Cityblock metric, also known as Manhattan distance or Taxicab metric, because the distance is computed by summing the distances of each vector component. A value of $p = 2$ is equal to the Euclidean distance and $p = \infty$ results in the Maximum distance, also known as Chebychev distance, computed as:

$$L_\infty(x, y) = max_{i=1}^{d} |x_i - y_i| \tag{5.8}$$

The following evaluation of distance metrics was done using seven different Minkowski metrics, for $p \in \{1, 1.2, 1.5, 1.7, 2, 2.2, 2.5\}$ as well as the cosine distance, which is measured by the cosine between the vectors x and y. The evaluation was done using three data sets – GTZAN, ISMIRgenre and ISMIRrhythm (see Chapter 4), as in Section 5.5 – and using the three feature sets (RH, RP and SSD) as well as four feature set combinations. The performance was measured by computing a distance matrix between all songs in a database and retrieving the five most similar songs to each song. The average percentage of retrieved songs having the same genre label as the query song is computed.

Diagrams from the evaluation are plotted in Figures 5.2 to 5.4. A line is depicted for each of the seven feature sets, the distance metrics used are given on the x-axis and the percentage of matching genres at the y-axis.

On both the GTZAN and ISMIRgenre data sets (Fig. 5.2 and 5.3, resp.) the SSD features achieved the best results. Combinations with other feature sets resulted in lower performance. Rhythm Histograms alone or in combination with Statistical Spectrum Descriptors achieved the lowest performance. The results on the ISMIRrhythm data set are completely contradictory – all *but* the SSD features performed very well. Oppositional performance on this database of some of the feature sets had been already observed in the experiments in Section 5.5. This behavior is probably due to the collection containing only dance rhythms of different styles with a distinct beat. As the Similarity task in MIREX 2006 was intended to evaluate similarity within a wide range of (particular popular) music, I did not further consider the results on the ISMIRrhythm data set. From the graphs of the two remaining collections the L_1-metric, i.e. the Cityblock metric, was identified to

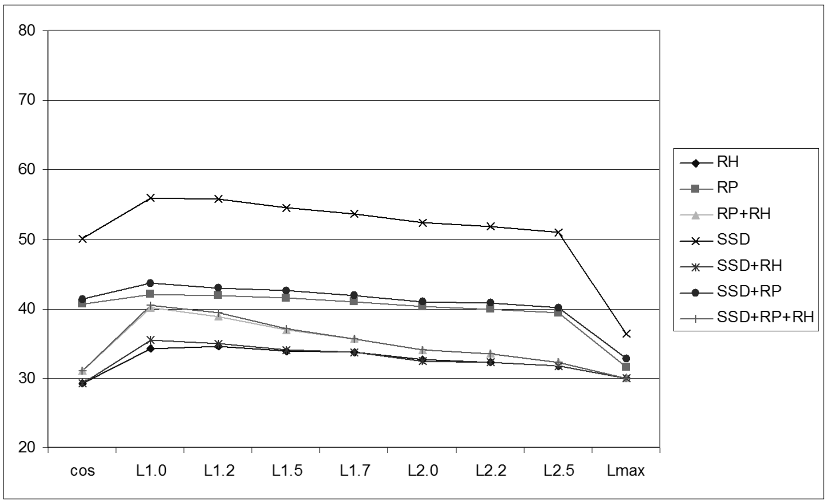


Figure 5.2: Distance metric evaluation. Percentage of matching genres for 5 similar songs, evaluated for 7 feature sets on the GTZAN collection.

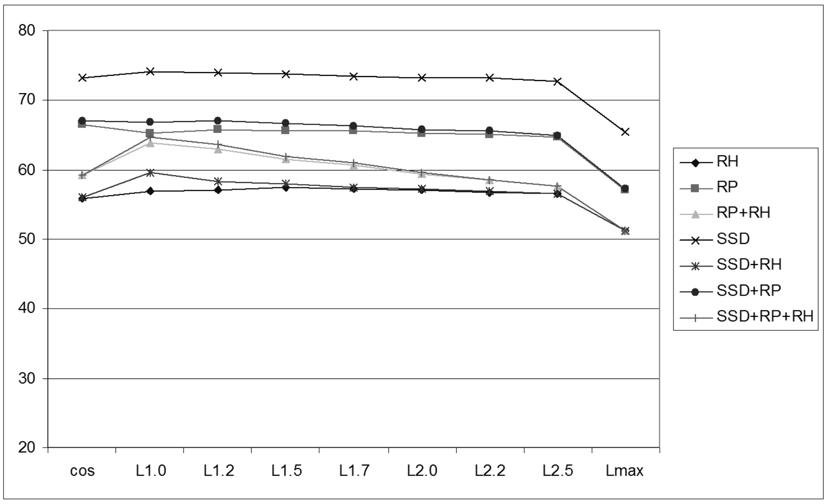


Figure 5.3: Distance metric evaluation. Percentage of matching genres for 5 similar songs, evaluated for 7 feature sets on the ISMIRgenre collection.

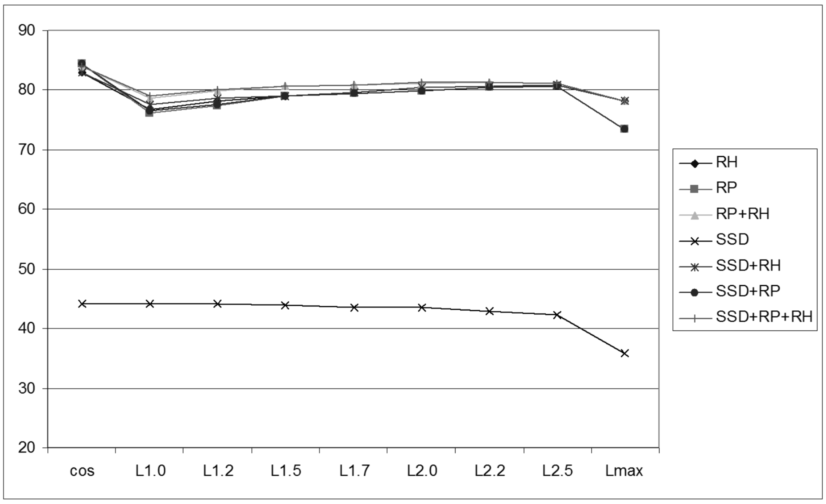


Figure 5.4: Distance metric evaluation. Percentage of matching genres for 5 similar songs, evaluated for 7 feature sets on the ISMIRrhythm collection.

deliver the best results on similarity retrieval with most of the feature sets, particularly using the SSD feature set, which showed the best performance. The positive implications for MIREX 2006 were, that a feature set that was computationally less costly than the other two and a distance metric which was simple to compute would deliver very reasonable results. These aspects were particularly important, as the computation times of the submitted algorithms would be considered in the MIREX 2006 evaluation and there were also restrictions on total runtime of the algorithms.

5.8 MIREX 2006

In MIREX 2006 the set of tasks had changed slightly, partly to follow the demands identified during MIREX 2005. For instance, instead of a Genre Classification task the "Audio Music Similarity and Retrieval" was introduced, comprising a human evaluation and an objective evaluation both based on a larger-scale music database containing 5000 pieces of audio. A similar strategy was pursued for Symbolic Similarity, hence the "Symbolic Melodic Similarity" task was created. Furthermore, a Query-by-Singing/Humming and a Score Following task had been introduced as well as an Audio Cover Song Identification task. This is the comprehensive MIREX 2006 task list:[23]

[23] http://www.music-ir.org/mirexwiki/index.php/Main_Page

- Audio Beat Tracking
- Audio Melody Extraction
- Audio Music Similarity and Retrieval
- Audio Cover Song Identification
- Audio Onset Detection
- Audio Tempo Extraction
- Query-by-Singing/Humming
- Score Following
- Symbolic Melodic Similarity

I participated in the MIREX 2006 Audio Music Similarity and Retrieval task and, with secondary interest, also in the Audio Cover Song Identification task. The submission to both tasks, described in the next section, was identical. In fact, submissions to the Audio Music Similarity and Retrieval task were evaluated automatically also on the Audio Cover Song Identification task, unless the participant disagreed.

5.8.1 Submitted Algorithm

An application that I was working on required the implementation of audio feature extractors in the Java programming language. The re-implementation of the Rhythm Patterns, Statistical Spectrum Descriptors and Rhythm Histograms feature sets was done by Simon Diesenreiter and me. The Java implementation has a number of advantages over the previous Matlab implementation, namely being more robust against errors, allowing the mixed usage of different audio formats and different sampling rates and recursion into arbitrary directory structures containing any number of audio files, among others. More interesting to this evaluation, however, was the fact that some parts of the feature extraction algorithms had to be implemented in slightly different ways, for instance using another library for FFT computation. Therefore I decided to participate with the new Java version in the large scale evaluation of MIREX 2006. As feature set, the SSD features have been chosen, according to the results of the pre-MIREX 2006 evaluation (see Section 5.7). The implementation of the SSD features followed largely the description of the algorithm in Section 3.2.6.

In MIREX 2006 audio files with 22.050 Hz sampling rate in mono format were provided. After segmentation of an audio file into segments of 2^{17} samples, the first and the last segment were skipped, from the remaining segments, every third one was processed. A feature vector was then calculated for each of the remaining segments.

First, the spectrogram was computed with an FFT using a Hanning window with a size of 512 samples and 50 % overlap. Then, the spectrum was aggregated to 23 critical bands according to the Bark scale. The Bark-scale spectrogram was then transformed into the decibel scale and subsequently into the Sone scale. From this representation of a segment's spectrogram seven statistical moments were computed per critical band, according to the description in Section 3.2.6, in order to describe fluctuations within the critical bands. The feature vector for an audio file was then constructed as the median of the SSD features of all extracted file segments.

For distance computation we submitted a script that loaded the feature vector file written by the Java SSD feature extractor software to Matlab, and computed a distance matrix from the feature vector space using the Cityblock metric (c.f. metric evaluation in Section 5.7).

5.8.2 Audio Music Similarity and Retrieval

The task was to submit an audio feature extraction algorithm and subsequently compute music similarity measures from which a distance matrix should be produced, i.e. a matrix containing the distances between all pairs of music tracks in a music database[24]. Feature extraction algorithms, any models and their parameters had to be trained and optimized in advance without the use of any data which has been part of the MIREX test database. The music database comprised 5000 pieces of (Western) music from 9 genres (see Table 4.5) in 22 kHz, mono, 16 bit Wave Audio format (including the tracks of the Audio Cover Song task - see Section 5.8.3 below). From the distance matrices, two forms of evaluations were performed: Evaluation based on human judgments and objective evaluation by statistic measures. Besides, the runtimes of the algorithms were recorded.

[24] http://www.music-ir.org/mirex2006/index.php/Audio_Music_Similarity_and_Retrieval

Human Evaluation

The primary evaluation focus of this MIREX 2006 task was on the judgements of the human evaluators. The human listening test was realized as follows:

60 songs were randomly selected as queries from the total of 5000 songs in the database. Each participating algorithm had to return the 5 most similar songs to the query (after filtering out the query itself, members of the cover song collection, as well as songs of the same artist as the query, in order to avoid the task to be an artist identification task). The results from all 6 participating algorithms then formed a list of 30 results per query, which had to be evaluated by human graders, who rated each retrieved song on two scales: one broad scale, stating whether the song is not, somewhat or very similar to the query song, and one fine-grained scale, where they had to score the retrieved songs on a real-value scale between 0 (not similar) and 10 (very similar). Each query/candidate list pair was evaluated by 3 different graders. 24 graders participated in the human evaluation, hence each person had to evaluate 7-8 query/candidate lists. The listening test was performed through the Evalutron 6000 web interface[25] created by the IMIRSEL team.

There were six participants in this task: Elias Pampalk (EP), Tim Pohle (TP), Vitor Soares (VS), Thomas Lidy & Andreas Rauber (LR), Kris West - Transcription model (KWT), Kris West - Likelihood model (KWL).

From the human judgments both the fine-grained score and the broad scale have been evaluated: The score for the fine-grained scale has been computed as the mean of all human ratings. For the broad scale, several different scoring systems have been applied, with different weighting of the 'very similar' and/or 'somewhat similar' grades. A table with all the scores for these different measures is available from the MIREX 2006 Audio Similarity and Retrieval results page[26]. The 6 different scoring systems resulted in a consistent ordering of the submitted algorithms, also the fine-grained and the broad scale results were consistent. A significance test has been applied to the results of the human evaluation in order to determine whether they indicate significant differences between the performance of the algorithms.

[25] http://www.music-ir.org/evaluation/eval6000/

[26] http://www.music-ir.org/mirex2006/index.php/Audio_Music_Similarity_and_Retrieval_Results

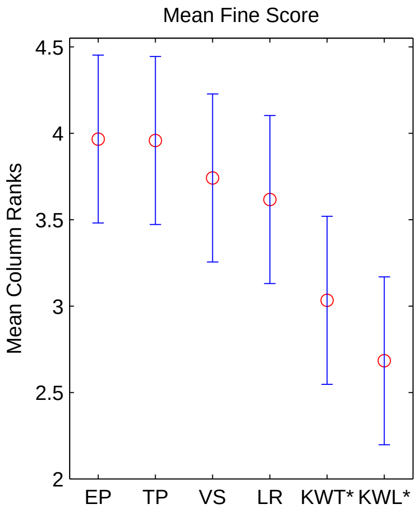


Figure 5.5: MIREX 2006 results from human listening tests, using the Friedman test. Circles mark the mean of the fine-grained human similarity scores, the lines depict the significance bounds at a level of p = 0.05.

The Friedman test [Fri37] was chosen because it is a non-parametric test which does not assume a normal distribution of the data. The Friedman test has been performed in Matlab with pairwise comparison of algorithms for each of the 60 queries, based on the fine-grained score. The results of the test at a confidence level of p = 0.05 showed that there are *no significant differences* between the top 5 algorithms (see Figure 5.5). Only the Likelihood algorithm by Kris West (KWL) performed significantly worse than three of the other algorithms. (The author however stated that there was a bug in his submissions.) As a consequence, there was no official ranking for this MIREX 2006 task.

Statistics

Computation of full distance matrices containing distances between all 5000 songs in the database enabled the computation of meta-data based statistics, such as: Average percentage of Genre, Artist and Album matches in the top 5, 10, 20 and 50 results, before and after artist filtering, Normalized average distance between examples of the same Genre, Artist or Album, Ratio of the average artist distance to the average genre distance, Number of times a song was similar to any of the 5000 queries, i.e. revealing songs that are always similar or never similar, Confusion Matrices, and more. One submission

(Vitor Soares, VS) has not been evaluated through these statistics, because the algorithm was not able to compute the full 5000 × 5000 distance matrix within the maximum time allowed for this MIREX 2006 task, which was 36 hours.

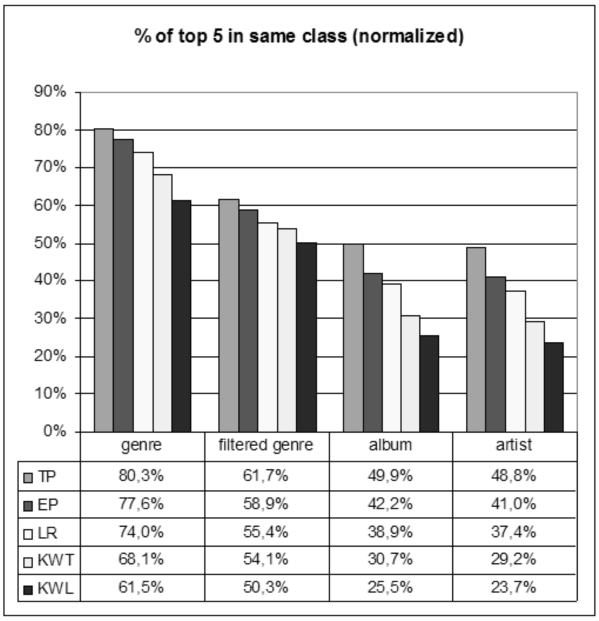
The results of this evaluation should be considered with caution, as the genre distribution in the music database was highly skewed: 50 % of the data was Rock music, 26.6 % Rap & Hip-Hop, 9.7 % Electronica & Dance, 5.3 % Country music and the remaining genres (Reggae, New Age, R & B, Latin and Jazz) were represented by 2 % or less, each. "Similar" songs, however, do not necessarily have the same genre label. This might be the reason why the ordering of the results from these statistics partly differs from the one of human listening results.

Figures 5.6 and 5.7 present the results of the percentages of how many within the retrieved 5 respectively 20 most similar songs had the same genre, artist or album as the query song. The numbers have been computed excluding the 330 cover songs and considering normalization for genres, artists or albums with less than 20 matches available in the database. The genre statistic is given before and after filtering out the query artist. The measurement of artist-filtered statistics is important, because many algorithms detect songs from the same artist as the most similar songs and unfiltered results evaluate mainly the capability of algorithms to identify artists. Further statistics for the top 10 and top 50 results are available from the Audio Music Similarity and Retrieval Statistics result web page[27]. In most of the cases our algorithm was ranked third, with a result of 74 % in a 5-nearest-neighbor-like genre recognition task. Considering the percentage of top 20 album matches our algorithm was ranked second (c.f. Figure 5.7). The changing order of result ranking seems to be an indication of the non-significant differences between the algorithms as revealed by the human evaluation.

Runtimes

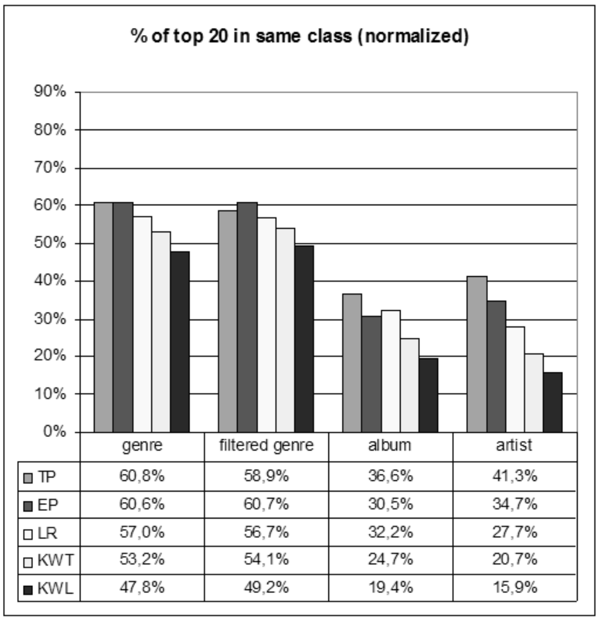
Computation times have been recorded individually for audio feature extraction and distance computation (except for the KWL model, where only the total time could be recorded). The runtimes were measured on Dual AMD Opteron 64 computers with 1.6 GHz and 4 GB RAM, running Linux

[27] http://www.music-ir.org/mirex2006/index.php/Audio_Music_Similarity_and_Retrieval_Other_Automatic_Evaluation_Results



	genre	filtered genre	album	artist
▨ TP	80,3%	61,7%	49,9%	48,8%
■ EP	77,6%	58,9%	42,2%	41,0%
□ LR	74,0%	55,4%	38,9%	37,4%
□ KWT	68,1%	54,1%	30,7%	29,2%
■ KWL	61,5%	50,3%	25,5%	23,7%

Figure 5.6: MIREX 2006 Audio Music Similarity and Retrieval: Average percentage of Genre (before and after artist filtering), Artist and Album matches in the *top 5* query results (normalized).



	genre	filtered genre	album	artist
▨ TP	60,8%	58,9%	36,6%	41,3%
■ EP	60,6%	60,7%	30,5%	34,7%
□ LR	57,0%	56,7%	32,2%	27,7%
□ KWT	53,2%	54,1%	24,7%	20,7%
■ KWL	47,8%	49,2%	19,4%	15,9%

Figure 5.7: MIREX 2006 Audio Music Similarity and Retrieval: Average percentage of Genre (before and after artist filtering), Artist and Album matches in the *top 20* query results (normalized).

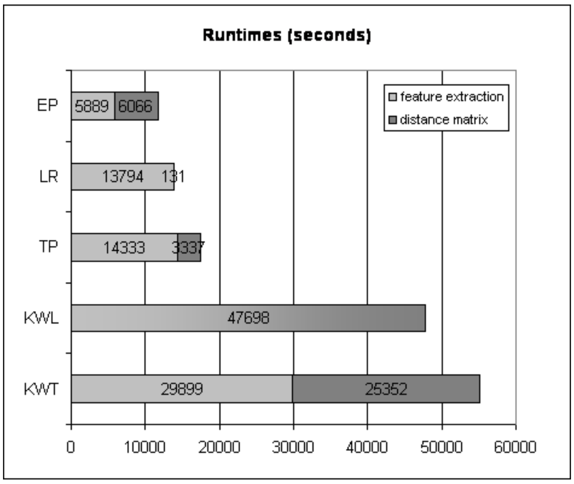


Figure 5.8: MIREX 2006: Runtimes of Audio Music Similarity algorithms in seconds (audio feature extraction and distance matrix computation).

(CentOS). The runtime of Soares' algorithm (VS) is not part of this comparison as it did not compute the full distance matrix. Pampalk's algorithm (EP) was the fastest in total (3 hours, 19 minutes) closely followed by ours (3 hours, 52 minutes) – c.f. Figure 5.8. Our algorithm was by far the fastest one in distance matrix computation (2 minutes only), which is due to the direct computation of distances in feature space using a simple distance metric, namely the Cityblock metric. Other algorithms needed a factor of 25 to 193 more time for distance computation. The total runtime of the slowest participating algorithm was about 4 times the runtime of ours.

5.8.3 Audio Cover Song Identification

The cover song database consisted of 30 different "cover songs" each represented by 11 different "versions", hence a total of 330 audio files. The cover songs represent a variety of genres (e.g., classical, jazz, gospel, rock, folk-rock, etc.) and the variations span a variety of styles and orchestrations.

Each of these cover song files has been used as a query and the top 10 returned items have been examined for the presence of the other 10 versions of the query file[28]. The 330 cover songs have been embedded within the 5000 songs database used for the Audio Music Similarity and Retrieval task which enabled an evaluation of the Similarity algorithms for the Cover Song task

[28] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

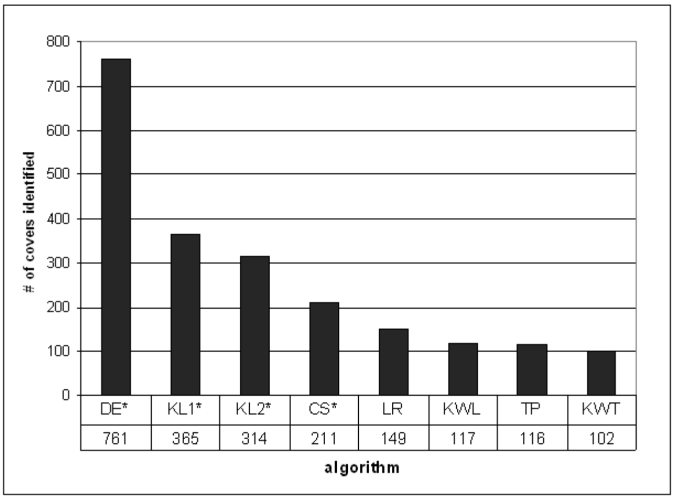


Figure 5.9: MIREX 2006 Audio Cover Song Identification results (total number of identified cover songs). Algorithms marked with * were specifically designed for the Cover Song Identification task.

without any extra effort except for retrieving the cover song queries from the distance matrices. For the evaluation of the Cover Song task, however, a reduced data set of 1000 songs has been used to accommodate more complex systems which have been particularly designed and submitted for cover song identification.

There were four submissions with systems which have been particularly designed for cover song identification – Dan Ellis (DE), Christian Sailer & Karin Dressler (CS), Kyogu Lee (KL, 2 models) – and four systems which have been evaluated as by-product of the Audio Music Similarity and Retrieval task (TP, LR, KWT and KWL – see Section 5.8.2).

The total number of correctly identified cover songs – out of the 3300 potentially detectable covers – is depicted in Figure 5.9. It can be seen from the results in the figure, that our submission was the best-performing "Audio Music Similarity and Retrieval" algorithm, outperformed however by the four specific cover song identification systems. Further measures – the mean number of covers identified, the mean of maxima (average of best-case performance) and the mean reciprocal rank of the first correctly identified cover (MRR) – are provided in a table on the Audio Cover Song Identification web page[29]. A Friedman test has been run against the MRR

[29] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song_Identification_Results

measure and identified Ellis' system (DE) as the clear winner of this task, while there was no significant difference between the 7 other algorithms.

5.8.4 Conclusions

The first large-scale human listening test for Music Similarity and Retrieval in MIREX showed, that our algorithms are competing with state-of-the-art algorithms – no significant difference in performance was determined between the top 5 algorithms. It is also one of the two fastest algorithms, with by far the most efficient distance calculation. Different statistics have been derived from genre, artist and album assignments, which gave our algorithm the third rank in most of the cases, and second rank in one case.

Our algorithm has also been evaluated on Audio Cover Song Identification together with three of the other Audio Music Similarity and Retrieval submissions and four submissions specifically designed for finding cover songs. It was the best on identifying covers out of the four Similarity algorithms, outperformed by the four specific Cover Song algorithms.

5.9 Conclusions

In this chapter a number of experiments were reported, aiming at evaluating and improving the performance of three different feature sets for MIR approaches. Most of the experiments were evaluated by the use of supervised classification, which enables quantitative performance measures of the employed approaches. Differences in classification results for various classifiers are dependent on the type and dimensionality of the feature set used. For feature sets with high dimension, such as for instance the Rhythm Patterns feature set, Support Vector Machines usually achieve better results than other classifiers.

Benchmarking in Music Information Retrieval has proven to be very important. Annual comparisons of state-of-the-art algorithms evaluate the progress and achievements that have been made and at the same time stimulate research for further improvements of the approaches.

An investigation of psycho-acoustic transformation steps within audio feature calculation has improved the performance of the Rhythm Patterns feature set. Two novel feature sets, as well as their combination, have been evaluated as being very competitive both on music genre classification and music similarity retrieval. The Rhythm Patterns feature set won the category of Rhythm Classification in the ISMIR 2004 Audio Description Contest.

Various distance metrics have been evaluated in experiments and eventually in the MIREX 2006 benchmark evaluation the presented approach, using Statistical Spectrum Descriptors, has been determined as being equally effective as four other top state-of-the-art algorithms, according to a significance test, and evaluated by a human listening test.

Chapter 6

Applications

6.1 Introduction

This chapter presents applications of content-based music descriptors (c.f. Chapter 3) for the efficient organization and visualization of music archives. Particularly, the Rhythm Patterns feature set is employed for the applications described in the following sections, but in principal every other feature set can be used. In 2001 the first map visualization of a music archive based on the Self-Organizing Map (SOM) has been presented [RF01]. The approach has been later extended to new visualizations [Pam01] and new interaction possibilities [NDR05]. The applications of these Music Maps are manifold. They can be used to represent music libraries graphically, to explore and browse music archives, to create playlist from ones personal music collection, to discover new music, etc.

After introducing the Self-Organizing Map algorithm in Section 6.2 a variety of different visualizations on top of such maps are reviewed in Section 6.3. Section 6.4 presents the interactive PlaySOM software, that allows to explore any kind of digital music archive and supports numerous interaction possibilities as well as the creation of playlists through trajectories of similar music. In Section 6.5 a lightweight application that is available for mobile devices such as PDAs and mobile phones is described. Eventually, in Section 6.6 an interesting example of the creation of a Music Map is shown, the analysis and the clustering of the complete works of Wolfgang Amadeus Mozart, which led to the creation of the Map of Mozart. Finally, in Section 6.7 conclusions of these application scenarios are given.

6.2 Self-Organizing Maps

There are numerous clustering algorithms that can be employed to organize audio by sound similarity based on a variety of feature sets. One model that is particularly suitable, is the Self-Organizing Map (SOM), an unsupervised neural network that provides a mapping from a high-dimensional input space to a usually two-dimensional output space [Koh01].

A SOM is initialized with an appropriate number i of units, proportional to the number of tracks in the music collection. Commonly, a rectangular map is chosen, but also toroidal maps are common, which avoid saturations at the borders of the map, but usually need unfolding for display on a 2D screen. With the MnemonicSOM [MMR05], the algorithm has been modified so that maps with virtually arbitrary shapes can be created. The i units are arranged on a two-dimensional grid. A weight vector $m_i \in \Re^n$ is attached to each unit. The input space is formed by the feature vectors $x \in \Re^n$ extracted from the music. Elements from the high-dimensional input space (i.e. the input vectors) are randomly presented to the SOM and the activation of each unit for the presented input vector is calculated using an activation function. The Euclidean distance between the weight vector of the unit and the input vector is commonly used for the activation function, nonetheless other distance functions can be employed. In the next step the weight vector of the unit showing the highest activation (i.e. having the smallest distance) is selected as the "winner" and is modified as to more closely resemble the presented input vector. The weight vector of the winner is moved towards the presented input signal by a certain fraction of the Euclidean distance as indicated by a time-decreasing learning rate α. Consequently, the next time the same input signal is presented, the unit's activation will be even higher. Furthermore, the weight vectors of units neighboring the winner are modified accordingly, yet to a smaller amount as compared to the winner. The magnitude of modification of the neighbors is described by a time-decreasing neighborhood function. This process is repeated for a large number of iterations, presenting each input vector in the input space multiple times to the SOM. The result of the SOM training procedure is a topologically ordered mapping of the presented input signals in the two-dimensional space. The SOM associates patterns in the input data with units on the grid, hence similarities present in the audio signals are reflected as faithfully as possible on the map, using the feature vectors extracted from audio.

The result is a similarity map, in which music is placed according to perceived similarity: Similar music is located close to each other, building clusters, while pieces with more distinct content are located farther away. If the pieces in the music collection are not from clearly distinguishable genres the map will reflect this by placing pieces along smooth transitions.

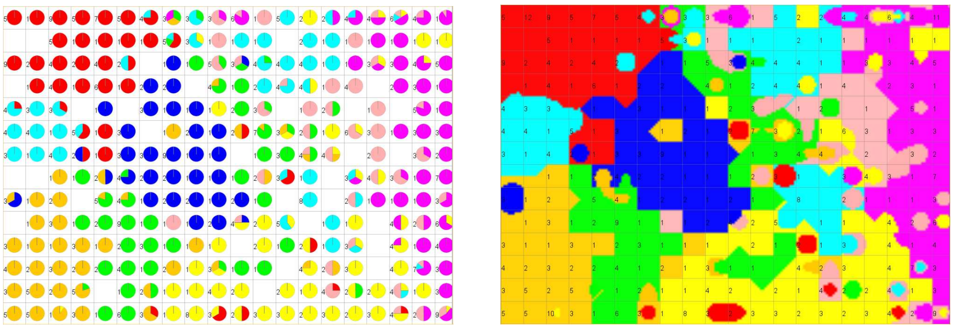
6.3 Visualizing Structures on the Self-Organizing Map

Due to the fact that the clusters and structures found by the trained music map are not inherently visible, several visualization techniques have been developed. Choosing appropriate visualization algorithms and appealing color palettes facilitate insight into the structures of the SOM from different perspectives. The influence of color palettes is important, as the different views on the data can be interpreted accordingly as mountains and valleys, islands in the sea, etc. For the following visualization examples the ISMIR-rhythm collection described in Section 4.2.3 was used to train a rectangular map with 20 x 14 units. As feature set for music similarity Rhythm Pattern features (see Section 3.2.5) have been used. The collection and hence the map contains music from eight different Latin American and Ballroom dances: ChaChaCha, Tango, Jive, Samba, Rumba, Quickstep, Slow Waltz, and Viennese Waltz. In order to elicit the cluster information a number of visualization techniques have been devised to analyze the map's structures. An additional visualization which is *not* based on the map structures but on external meta-data is the class visualization. In order to give an overview of the music map used in the examples the class visualization will be described first. An interactive web demo of a music map trained from the same music collection is available in the Web[1].

Class Visualization

While the Self-Organizing Map does not rely on any manual classification, nevertheless there are often genre labels available for the music titles (or at

[1] http://www.ifs.tuwien.ac.at/mir/playsom/demo/



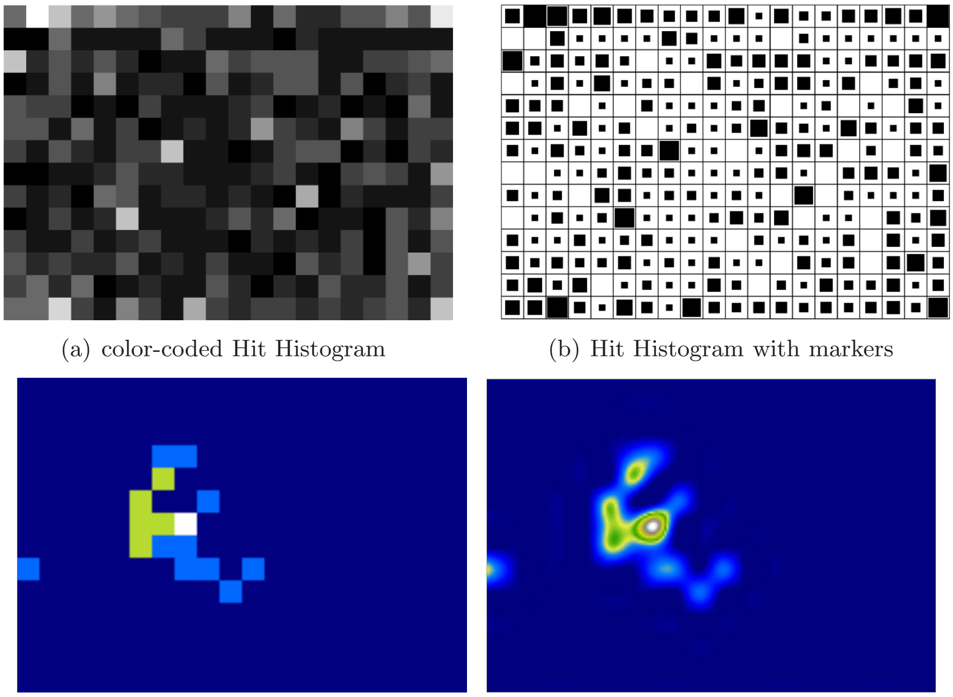
(a) Pie chart overlay

(b) Voronoi-like tessellation

Figure 6.1: Class Visualization. Classes: ChaChaCha (red), Tango (cyan), Jive (blue), Samba (orange), Rumba (yellow), Quickstep (green), SlowWaltz (pink), VienneseWaltz (magenta)

least for the artists). Also, music collections sometimes have been manually sorted into different classes. The availability of genre or class information allows the creation of a visualization which assists with the analysis of the clustered structures on the map. External genre information allows to color-code the genres and to overlay the genre information onto other existing visualizations. Commonly, pie-charts (c.f. Figure 6.1(a)) are used to visualize the distribution of the classes within a map unit, thus, one pie-chart is placed on every unit of the map. Alternatively, if a full-area visualization is preferred, a Voronoi-like approach (c.f. Figure 6.1(b)) is taken to fill regions covered by a single genre with its respective color. If units contain a mixed set of classes, an approach similar to dithering is used to represent multiple classes within the area covered by that unit [Azi06]. The additional clues provided by genre or class visualization vastly facilitate the description of regions or clusters identified by other visualizations, such as Smoothed Data Histograms.

Describing the map (c.f. Figure 6.1), the cluster on the top left contains ChaChaCha, Samba was clustered bottom left, with Quickstep just to its right. Jive music is found in a cluster slightly left of the center, Rumba at the bottom, Slow Waltz at the outer right, with Viennese Waltz to its left. Tango has been partitioned into three clusters, one at the center left, one at the top and another part below Viennese Waltz. Parts of Quickstep are also found left of the latter two Tango clusters.



(a) color-coded Hit Histogram

(b) Hit Histogram with markers

(c) Hit Histogram of a query for *Jive* music (flat)

(d) Hit Histogram of a query for *Jive* music (spline interpolated)

Figure 6.2: Hit Histograms

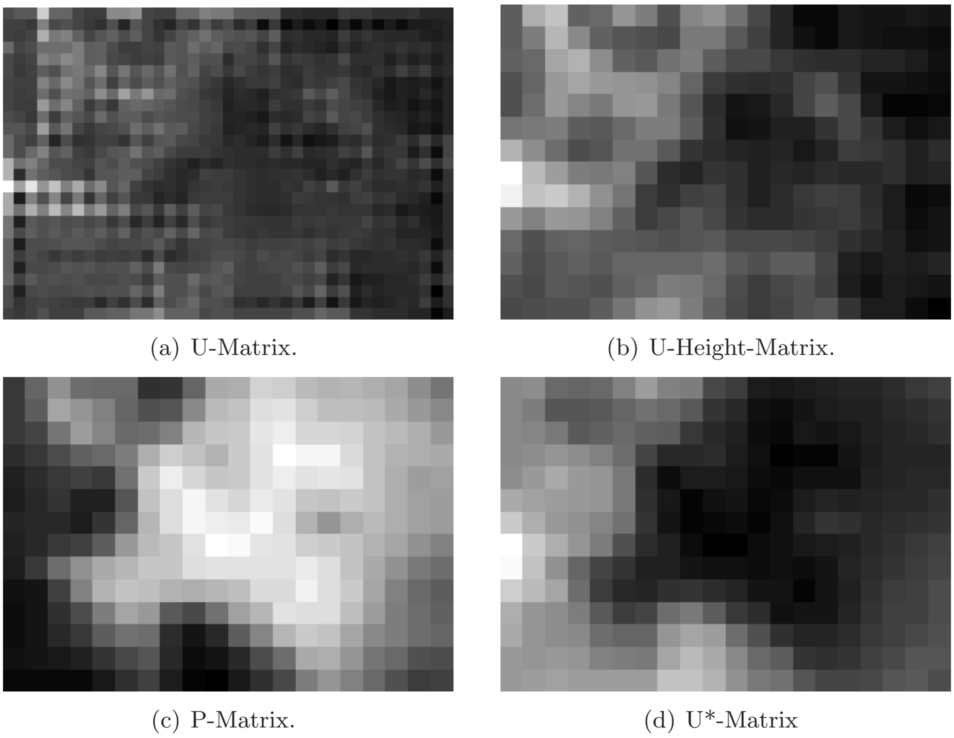
Hit Histograms

A Hit Histogram visualization depicts the distribution of data over the SOM. For each unit the number of items (songs) mapped is counted to compute the Hit Histogram. Different visualizations of Hit Histograms are possible: Likewise to other visualizations, the number of mapped songs can be visualized by different colors (Figure 6.2(a)). Another variant is the use of markers (circles, bars, etc.) on top of the map grid, where the number of hits for each unit is reflected by the size of the marker (Figure 6.2(b)). Hit Histograms are good for getting an overview of the map and an indication where much of the data is concentrated. In Figures 6.2(a) and 6.2(b) for example the highest peaks correspond to the ChaChaCha, Samba and Slow Waltz clusters, while the visualization also exhibits lower peaks for other clusters.

The approach is also very useful if not the entire data set is to be visualized, but only a part of it. With the same techniques it is possible to display (statistical) information about a sub-set of the data collection. This is used, for instance, to visualize results of a query to the music map: Querying the map with the name of a certain artist, a Hit Histogram is constructed by counting the number of times this artist is present with a piece of music on every unit. Only a part of the map will have hits on that query, the histogram values of the respective units are increased by one for each hit, while the values of the remaining units are set to zero. The visualization provides an immediate overview of where the resulting items of the query are located. Depending on the graphical result, one receives an indication of whether the music of a given artist is rather distributed or aggregated in a certain area of the map. Hit Histograms can be employed in a number of situations: If genre or class labels of the songs are available, the distribution of a particular genre can be visualized with Hit Histograms. Figure 6.2(c) shows a Hit Histogram of the distribution of "Jive" music within the example music collection in a color-coded representation similar to the one in Figure 6.2(a). In Figure 6.2(d) spline interpolation has been used in order to improve the intuitive perception of clusters in the query results. Another possibility would be the visualization of cover songs, having the same title, but different performers. Hit Histograms may be a useful visualization for virtually any other sort of query where frequency counts are involved.

U-Matrix

One of the first and most prominent SOM visualizations developed is the U-Matrix [US90]. The U-Matrix visualizes distances between the weight vectors of adjacent units. The local distances are mapped onto a color palette: small distances between neighboring units are depicted with another color than large distances, the color is gradually changing with distance. As a consequence the U-Matrix reveals homogeneous clusters as areas with one particular color, while the cluster boundaries, having larger distances, are visualized with another color. In Figure 6.3(a) the cluster boundaries are shown by bright colors. There is e.g. a particularly strong boundary between Tango and Samba music as well as between Quickstep and Rumba. With an appropriate color palette one can give this visualization the metaphor of mountains and valleys: Large distances are visualized with brown or even white color (for the mountain tops), lower distances are visualized with different shadings of green color. The mountains then indicate the barriers



(a) U-Matrix.

(b) U-Height-Matrix.

(c) P-Matrix.

(d) U*-Matrix

Figure 6.3: U-Matrix, U-Height-Matrix, P-Matrix and U*-Matrix

between homogeneous (flat) regions. As the U-Matrix is computed from distances between adjacent units, the visualization result is depicted at a finer level as other (per-unit) visualizations. The U-Height-Matrix depicted in Figure 6.3(b) is a unit-wise aggregation of adjacent U-Matrix values.

P-Matrix

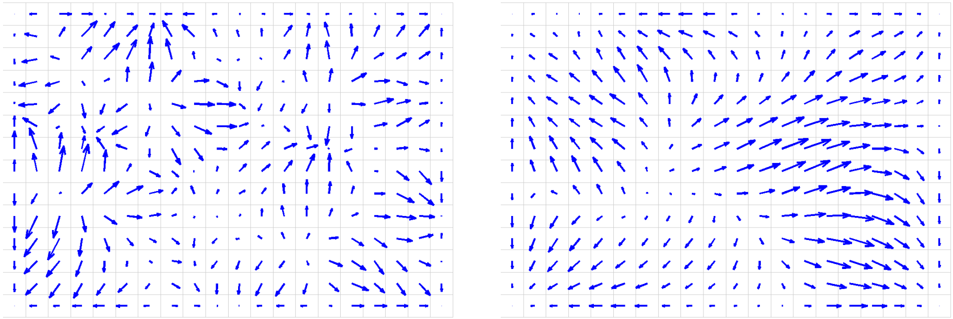
The P-Matrix visualization shows local relative data densities based on an estimated radius around the unit prototype vectors of the SOM. First, the so-called Pareto-Radius is determined as a quantile of the pair-wise distances between the data vectors [Ult03a]. Then, for each map unit, the number of data points within the sphere of the previously calculated radius is counted and visualized on the map grid. The purpose of this visualization is to show the relative density of the map units. The map nodes in the center of the map usually have a higher P value than the ones at the border, which is also the case in Figure 6.3(c).

U*-Matrix

The U*-Matrix aims at showing cluster boundaries taking both the local distances between the unit vectors and the data density into account [Ult03b]. It is derived from the P-Matrix and the U-Matrix. This is performed by weighting the U-Matrix values according to the P-Matrix values: Local distances within a cluster with high density (high P-Matrix values) are weighted less than distances in areas with low density, resulting in a smoother version of the U-Matrix. The intention is to reduce the effects of inhomogeneous visualization in actually dense regions. Comparing Figures 6.3(b) to 6.3(d)), it can be seen that the dense region exhibited in the P-Matrix is visualized more homogeneously in the U*-Matrix visualization than in the U-Matrix. Consequently, the smaller distances in the dense areas have vanished while the cluster boundaries between Tango and Samba as well as Quickstep and Rumba are depicted more clearly. The U*-Matrix has been particularly designed for Emergent SOMs [Ult99], i.e. very large SOMs, where the number of units on the map is much larger than the number of input data.

Gradient Fields

The Gradient Field visualization [PDR06] aims at making the SOM readable for persons with engineering background who have experience with flow and gradient visualizations. It is displayed as a vector field overlay on top of the map. The information communicated through the gradient field visualization is similar to the U-Matrix, identifying clusters and coherent areas on the map, but allowing for extending the neighborhood width, and thus showing more global distances. Another goal is to make explicit the direction of the most similar cluster center, represented by arrows pointing to this center. The method turns out to be very useful for SOMs with a large numbers of map units. The neighborhood radius is an adjustable parameter: a higher radius has a smoothing effect, emphasizing the global structures over local ones. Thus, this parameter is selected depending on the level of detail one is interested in. Figure 6.4(a) depicts a gradient field with a neighborhood radius of 2, with the arrows indicating the direction to the center of each genre cluster (compare Figure 6.1). In Figure 6.4(b) the parameter was set to 7, with the result of the arrows pointing mostly to the most salient cluster peaks exhibited in the Hit Histogram (c.f. Figures 6.2(a) and 6.2(b)).



(a) $\sigma = 2$

(b) $\sigma = 7$

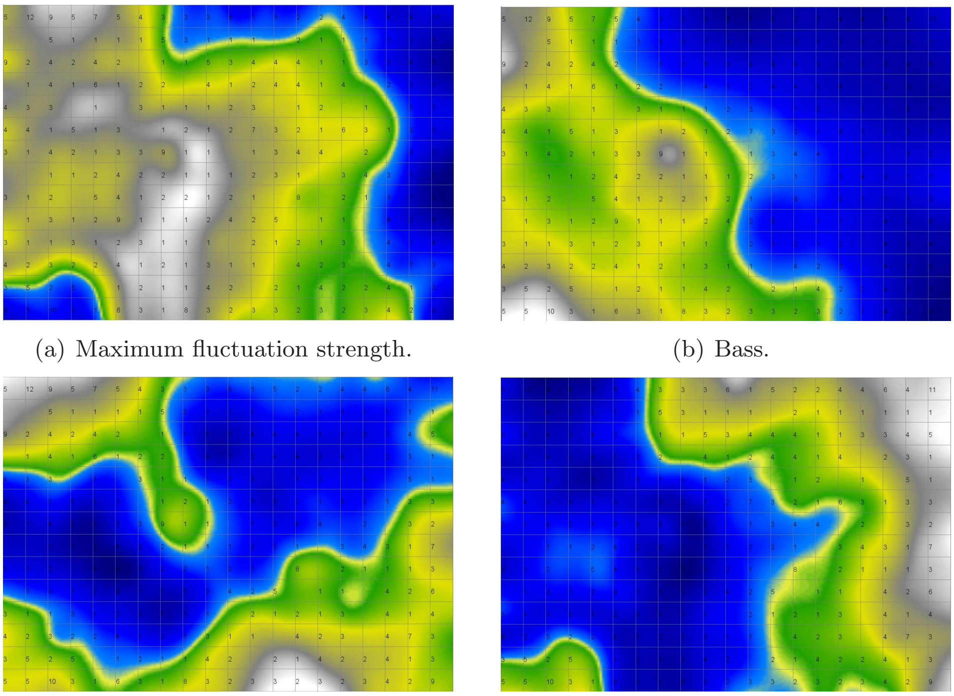
Figure 6.4: Gradient Field visualization with neighborhood parameter σ set to different values

Component Planes

Component planes visualize the distribution of particular features or attributes (components) of the feature set. A single component of the unit weight vectors is used to create the visualization allowing to investigate the influence of a particular feature (such as the Zero Crossing Rate or a modulation frequency on a specific band within the Rhythm Patterns feature set) to the mapping of certain pieces of music on particular regions of the map. For the component planes visualization, each unit on the map is color-coded, where the color reflects the magnitude of a particular component of the weight vector of each unit. With the appropriate color palette, this visualization is comparable to "Weather Charts" [PRM02].

When maps are created using feature sets with large dimensions, the visualization of every component of the feature set is probably not desired. Especially feature sets that allow the aggregation of attributes to semantic sub-sets are suitable to create a "Weather Chart"-like visualization, permitting the description of map regions by comprehensive terms. For this purpose sub-sets of feature vector components are being accumulated. Particularly for the Rhythm Patterns feature set (c.f. Section 3.2.5) four "Weather Chart" visualizations have been created reflecting the psycho-acoustic characteristics inherent in the feature set:

Maximum fluctuation strength is calculated as the highest value contained in the Rhythm Pattern. Its Weather Chart indicates regions with music dominated by strong beats.



(a) Maximum fluctuation strength.

(b) Bass.

(c) Non-aggressiveness.

(d) Low frequencies dominant.

Figure 6.5: Component Planes: Weather Chart visualizations of characteristics inherent in the Rhythm Patterns feature set

Bass denotes the aggregation of the values in the lowest two critical bands with a modulation frequency higher than 1Hz indicating music with bass beats faster than 60 beats per minute.

Non-aggressiveness takes into account values with a modulation frequency lower than 0.5 Hz of all critical bands except the lowest two. The respective Weather Chart indicates rather calm songs with slow rhythm.

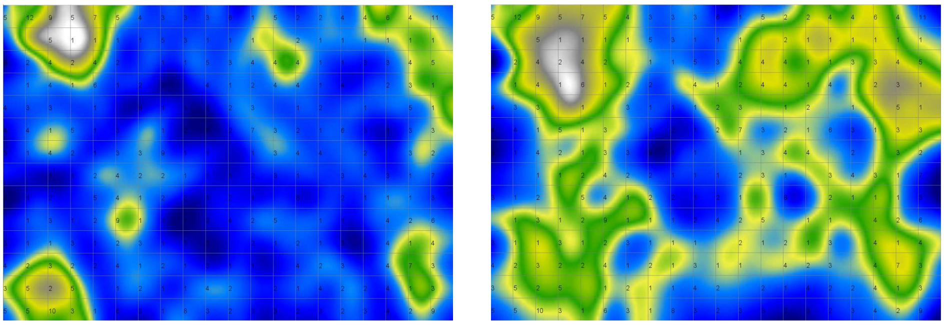
Low frequencies dominant is the ratio of the five lowest and highest critical bands and measures in how far the low frequencies dominate.

As these examples show, Component Planes provide an intuitive explanation of the map, its regions and the underlying features. Figure 6.5 shows examples of the four Rhythm Patterns Weather Charts visualizations explained. Regarding the figures we see that the maximum magnitude of fluctuation strength corresponds to Quickstep and Jive music. Bass covers the genres ChaChaCha, Jive, Samba, Quickstep and Rumba and partly Tango. ChaChaCha and Rumba have been identified to have the least aggressive rhythm, while in Slow Waltz and Viennese Waltz the low frequencies dominate.

Smoothed Data Histograms

Detecting and visualizing the actual cluster structure of a map is a challenging problem. The U-matrix described above visualizes the distances between the model vectors of units which are immediate neighbors, aiming at cluster boundary detection. Smoothed Data Histograms [RPM03] are an approach to visualize the cluster structure of the data set in a more global manner. The concept of this visualization technique is basically a density estimation and resembles the probability density of the whole data set on the map. When a SOM is trained, each data item is assigned to the map unit which best represents it, i.e. the unit which has the smallest distance between its model vector and the respective feature vector. However, by continuation of these distance calculations it is also possible to determine the second best, third best, and so on, matching units for a given feature vector. A voting function is introduced using a robust ranking, which assigns points to each map unit: For every data item, the best matching unit gets n points, the second best $n-1$ points, the third $n-2$ and so forth, for the n closest map units, where n is the user-adjustable smoothing parameter. All votes are accumulated resulting in a histogram over the entire map. The histogram is then visualized using spline interpolation and appropriate color palettes. Depending on the palette used, map units in the centers of clusters are represented by mountain peaks while map units located between clusters are represented as valleys. Using another palette the SDH visualization creates the *Islands of Music* [Pam01] metaphor, ranging from dark blue (deep sea), via light blue (shallow water), yellow (beach), dark green (forest), light green (hills), to gray (rocks) and finally white (snow).

The SDH visualization, contrary to the U-Matrix, offers a sort of hierarchical representation of the cluster structures on the map. On a higher level the overall structure of the music archive is represented by large continents or islands. These larger genres or styles of music might be connected through land passages or might be completely isolated by the sea. On lower levels the structure is represented by mountains and hills, which can be con-



(a) smoothing factor $n = 12$

(b) smoothing factor $n = 33$

Figure 6.6: Smoothed Data Histograms

nected through a ridge or separated by valleys. For example, there might be an island (or even a "continent") comprising non-aggressive, calm music without strong beats. On this island there might be two mountains, one representing classical and the other one orchestral film music, which is somewhat more dynamic. Another example might be an island comprising electronic music and the hills and mountains on it representing sub-genres with different rhythm or beat. This does not imply that the most interesting pieces are always located on or around mountains, interesting pieces might also be located between two strongly represented distinctive groups of music, and would thus be found either in the valleys between mountains or even in the sea between islands, in the case of pieces which are not typical members of the main genres or music styles represented by the large islands (clusters) on the map.

The parameter n mentioned before, which determines the number of best matching units to be considered in the voting scheme for the SDH, can be adjusted by the user to interactively change the appearance of the SDH visualization. A low value of n creates more and smaller clusters (islands) on the map, with an increasing value of n the islands grow and eventually merge building greater islands or continents. Analogous to the hierarchical representation described before, this enables the user of the map to create a cluster structure visualization at different levels, depending if a more general aggregation of the data or a more specialized one is desired.

Figure 6.6 shows two SDH visualizations of the ISMIRrhythm music collection described at the beginning of this section (and in more detail in Section 4.2.3): in the left visualization, the smoothing parameter was set to 12, showing the dominant clusters of ChaChaCha, Samba and Slow Waltz, as well as the peaks of Tango and Quickstep. On the right image the smoothing parameter was set to 33, showing large clusters which are beginning to merge, joining also the two parts of the ChaChaCha cluster and the Slow Waltz clusters which were separated in Figure 6.6(a). Moreover, Jive music is found by a "sea-ground level" cluster in the center of the map, surrounded by "islands". Jive was the genre with the lowest number of pieces in the collection which is probably the reason why the SDH visualization does not show an "island" cluster of Jive as well.

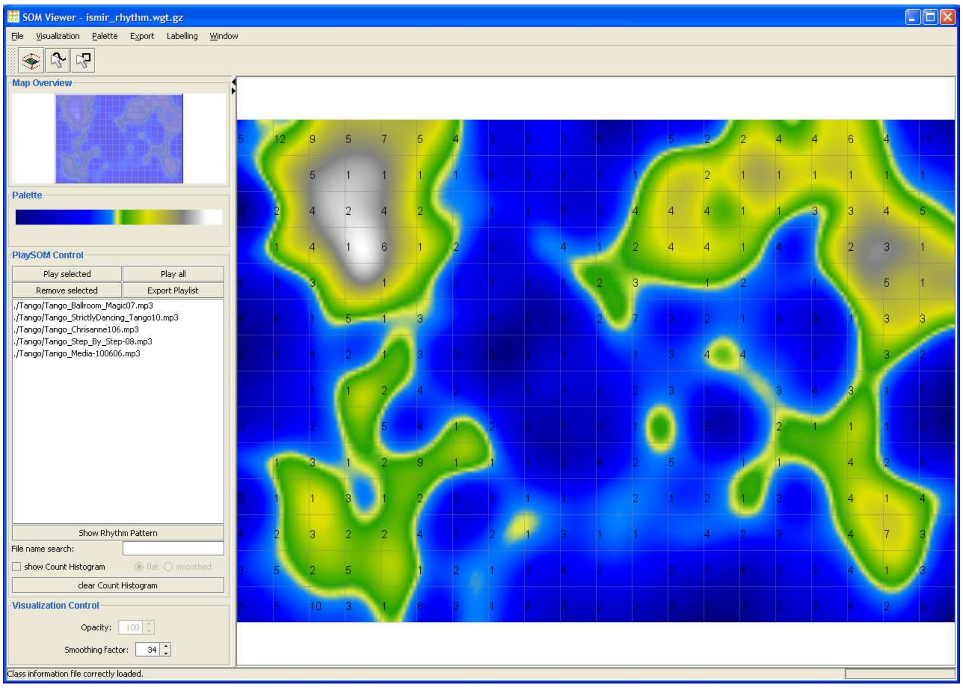
6.4 PlaySOM – Interaction with Music Maps

Music Maps provide a convenient overview of the content of music archives. Yet, their advantages are augmented by the PlaySOM application, which enriches music maps with facilities for interaction, intuitive browsing and exploration, semantic zooming, panning and playlist creation. This moves the SOM from a purely analytical machine learning tool for analyzing the high-dimensional feature space to an actual and direct application platform. PlaySOM is based on the SOMViewer application originally developed by Michael Dittenbach. Lateron, new interaction models [NDR05], new visualizations [Azi06] (among others) and a modified SOM algorithm (MnemonicSOM, [MMR05]) were added step by step. My own contributions include a new query interface as well as the spline-interpolated query-result Hit Histograms described in Section 6.3.

Interface

The main PlaySOM interface is shown in Figure 6.7. Its largest part is covered by the interactive map on the right, where squares represent single units of the SOM. At the outmost zooming level, the units are labeled with numbers indicating the quantity of songs per unit. The left hand side of the user interface contains

- a birds-eye-view showing which part of the potentially very large map is currently displayed in the main view
- the color palette used in the currently active visualization



./Tango/Tango_Ballroom_Magic07.mp3
./Tango/Tango_StrictlyDancing_Tango10.mp3
./Tango/Tango_Chrisanne106.mp3
./Tango/Tango_Step_By_Step-08.mp3
./Tango/Tango_Media-100606.mp3

File name search:

□ show Count Histogram ● flat ○ smoothed

Opacity: 100

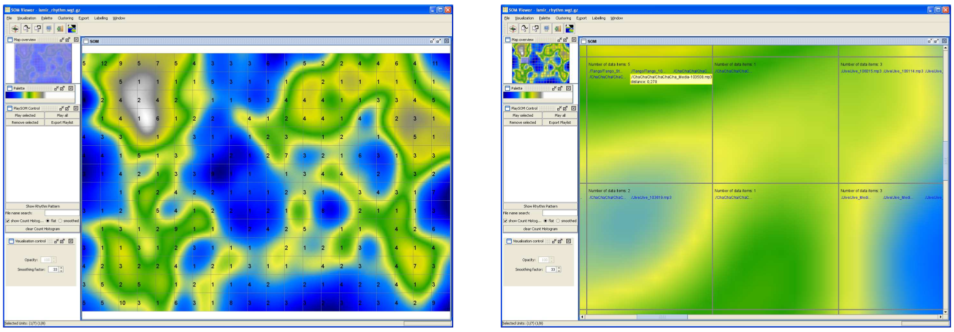
Smoothing factor: 34

Class information file correctly loaded.

Figure 6.7: PlaySOM desktop application: main interface

- the playlist containing titles of the last selection made on the map, alongside with buttons to refine, play or export the playlist
- search fields for queries to the map
- a visualization control to influence the parameters of the currently active visualization

The menu bar at the top of the window contains menus for additional settings, for switching between the visualization, changing the palette and for exporting the map into different formats, including the PocketSOMPlayer format. Also, if genre tags are available for the music titles, the distribution of genres on the music map can be displayed as colored overlay as an additional clue. A toolbar allows the user to switch between the two different selection models and to automatically zoom out to fit the map to the current screen size.



(a) Low zooming level: the number of songs mapped on the units are indicated.

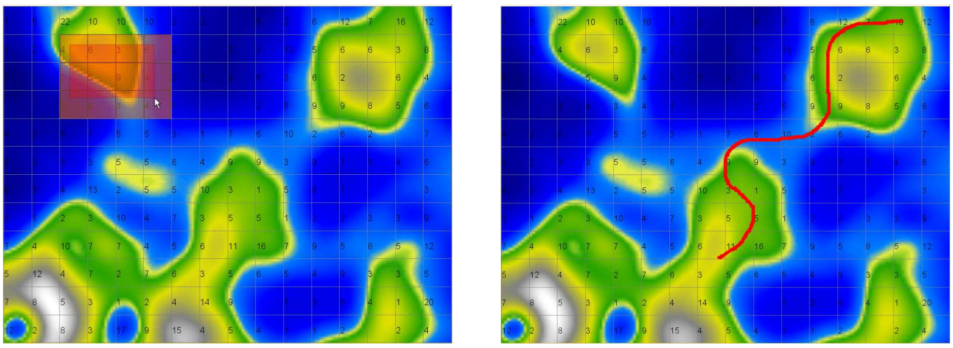
(b) High zooming level: song titles and additional information are displayed on the respective units.

Figure 6.8: Semantic zooming and its influence on the amount of information displayed

Interaction

PlaySOM allows the user to select from and to switch between the different visualizations described in the previous section. The Weather Charts visualization for instance, indicating particular musical attributes (see Section 6.3), aids the user in finding the music of a particular genre or style. With the SDH visualization creating an *Islands of Music* interface a metaphor for a geographic map is offered, which allows for intuitive interaction with the music map. Users can move across the map, zoom into areas of interest and select songs they want to listen to. With increasing level of zoom the amount and type of data displayed is changed (c.f. Figure 6.8), providing more details about the items on the units. The interaction model allows to conveniently traverse and explore the music map. At any level of detail users can select single songs and play them, or create playlists directly by selection on the map. Playlists can either be played immediately or exported for later use.

The application also offers traditional search by artist name or song title, and locates the retrieved titles on the map by marking the respective units with a different color. Alternatively, the results of a query are visualized using Hit Histograms (see Section 6.3), showing the distribution of the search results on the map including the number of hits per unit. From the retrieved locations it is easy to browse for and to discover similar (yet unknown) music simply by selecting the SOM units close to the marked location.



(a) Rectangle selection model.

(b) Trajectory selection model.

Figure 6.9: Different models of selecting music on the PlaySOM music map

Playlist Creation

The two selection models offered by PlaySOM allow the user to directly create playlists by interacting with the map. Thus the application not only allows for convenient browsing of music collections containing hundreds or thousands of songs, it also enables the creation of playlists based on real music similarity instead of albums or meta-data. This relieves users from traditional browsing through lists and hierarchies of genres and albums, which often leads to rather monotonous playlists consisting of complete albums from a single artist. Instead of the burdensome compilation of playlists title by title, PlaySOM allows to directly select a region of the map with the music style of interest. Moreover, by drawing trajectories on the map, it is possible to generate playlists which are traversing multiple genres. Figure 6.9 depicts the playlist creation models that are supported by PlaySOM. The rectangular selection model (c.f. Figure 6.9(a)) allows the user to drag a rectangle and select the songs belonging to units inside that rectangle without preserving any order of the selected tracks. This model is used to select music from one particular cluster or region on the map and is useful if music from a particular genre or sub-genre well-represented by a cluster is desired. The path selection model allows users to draw trajectories and select all songs belonging to units beneath that trajectory. Figure 6.9(b) shows a trajectory that moves from one music cluster to another one, including the music that is located on the transition between those clusters. Paths can be drawn on the map, for instance, starting with Slow Waltz music going

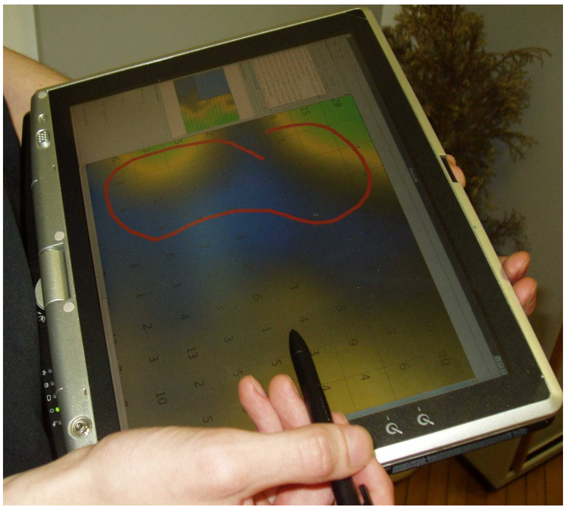


Figure 6.10: PlaySOM running on a Tablet PC with pen input

via Tango to Jive music and back to Slow Waltz via Viennese Waltz. It can be fixed from the beginning that such a "tour" should take, for example, two hours. The PlaySOM application then automatically selects music along these path lines, or, optionally, plays music randomly from within the trajectory drawn. Such an approach offers a wonderful possibility to quickly prepare a playlist for particular situations (party, dinner, background music, etc.). Once a user has selected songs on the map the playlist element in the interface displays the list of selected titles. It is possible to play the music in the list directly or to refine the list by manually dropping single songs from the selection. The playlist can also be exported for later use on the desktop computer or on other devices like mobile phones, PDAs or audio players. The music can be either played locally or, if the music collection is stored on a server, via a streaming environment. Furthermore, the PlaySOM application can be conveniently and efficiently used on a Tablet PC (see Figure 6.10), because its interface is easily controllable via pen input. It is even usable as a touch screen application.

Summarizing, the PlaySOM interface allows the interactive exploration of entire music archives and the creation of personal playlists directly by selecting regions of one's personal taste, without having to browse a list of available titles and manually sorting them into playlists. Thus, the map metaphor constitutes a completely novel experience of music retrieval by navigation through "music spaces".

6.5 PocketSOMPlayer – Music Maps on Mobile Devices

Traditional selection methods such as browsing long lists of music titles or selecting artists from alphabetical lists or entering queries into a search field are particularly cumbersome when used on mobile devices, such as PDAs, mobile phones or portable audio players. Yet, this issue becomes even more annoying with people's music collections constantly getting larger.

The need for improved access to music collections on portable devices motivated the implementation of music maps on those devices, allowing for direct and intuitive access to the desired music. Like for PlaySOM a Self-Organizing Map builds the basis for creating intuitive visualizations and forms the application interface. A lightweight application has been created that runs on Java-enabled PDAs and mobile phones (see Figure 6.11). The application takes an image export (e.g. an Islands of Music (SDH) visualization) from the PlaySOM application for its interface, i.e. currently the PlaySOM application is needed to create a map for the PocketSOMPlayer.

Interaction

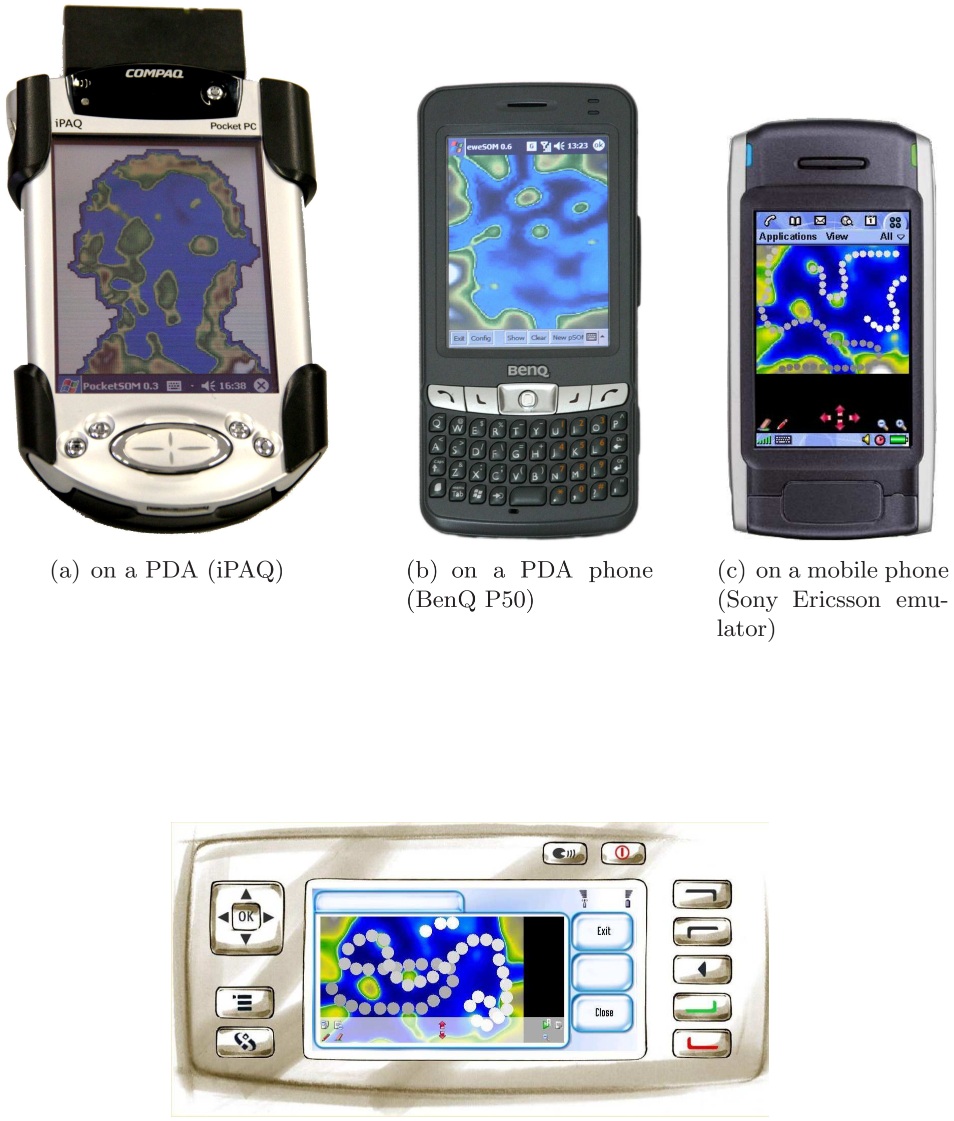
The interface offered by the PocketSOMPlayer [NDR05] is similar to its desktop counterpart PlaySOM. It also offers exploring a music map by zooming and selection and playlists are created by drawing paths with a pen on the screen (provided the device supports pen input). Due to the limitations in screen size, playlists are displayed on the full screen after a selection was made, offering the choice of fine-tuning the playlist.

Playing Scenarios

Several play modes exist for the PocketSOMPlayer:

First, if the device has sufficient capacity to store entire music collections on it, music can be played directly from the device.

Alternatively, the PocketSOMPlayer can also be used for streaming one's personal music collection from the desktop computer at home. A connection is opened from the mobile device to one's personal computer and each time a playlist is created by drawing a path on the mobile device, the PocketSOMPlayer starts to stream the music from the desktop computer to the



(a) on a PDA (iPAQ)

(b) on a PDA phone (BenQ P50)

(c) on a mobile phone (Sony Ericsson emulator)

(d) on a multimedia phone (Nokia 7710 emulator)

Figure 6.11: Different implementations of the PocketSOMPlayer

handheld device.

Instead of streaming the music to the mobile device, the PocketSOMPlayer can be used also as convenient remote control to select music one wants to listen to in one's living room. After selecting a path or an area on the music map on the PDA or mobile phone a playlist is sent to the desktop computer which then plays the music.

With an active connection to the Internet the PocketSOMPlayer is able to stream the music on the selected map trajectory from a server. Thus, while traveling around, with this technology one can access a music repository from wherever one has access to the Internet, be it via GPRS, UMTS or Wireless LAN. This enables also the idea of a central music repository with a huge archive of music in it and a multitude of users accessing this music from wherever they are, offering room for portal-based service providers.

Conclusion

Selecting music via drawing trajectories on a touch screen is straightforward, easy to learn and intuitive as opposed to clicking through hierarchies of genres or interprets. The PocketSOMPlayer offers a convenient alternative to traditional music selection and may also constitute a new model of how to access a music collection on portable audio players.

6.6 The Map of Mozart

The 250th anniversary of Wolfgang Amadeus Mozart in 2006 was the motivation to acquire the collection of his complete works and to analyze all the pieces of music that W. A. Mozart ever created by content-based audio feature extraction. This collection of 2442 pieces of music by a single composer from a specific period of time is characterized by being very homogeneous, nevertheless the music can be divided into a set of categories, such as symphonies, serenades, sacred works, violin sonatas, operas, etc. (17 classes in total, see Table 4.6, which also includes the number of works in each category). This categorization is based partly on the covers of the original CDs and some categories have been manually further subdivided, where it made sense.

The Map of Mozart clusters the entire Mozart collection without the use of any genre information, solely based on the automatically extracted audio features. The genre information can be later overlaid on the Map of Mozart as a visual hint to evaluate the clustering of the music.

Feature Extraction

For feature extraction from the digital audio, the Rhythm Patterns feature set (see Section 3.2.5) has been employed. The standard algorithm (in the Matlab version) which includes psycho-acoustic processing has been used, however without considering Spectral Masking and without the subsequent filtering and smoothing step. Every fourth 6 second segment of the pieces has been considered, without the first and the last segment of a piece of music. Capturing fluctuations on all human audible frequency regions, the features are capable not only to discover rhythmics, but also timbral features and thus are able to recognize different instrumentation in music.

Clustering

In the subsequent step, the features extracted have been used for input to clustering using a Self-Organizing Map. A rectangular map might be sub-optimal for memorizing the orientation of a map, i.e. the location of different types of music on the map, and for explaining to people where genre-like clusters are located on the map. Therefore, a modified SOM algorithm has been used, the so-called Mnemonic SOM [MMR05]. Instead of creating rectangular maps this novel SOM method enables the use of memorizable shapes, e.g. in the form of countries, geometrical figures,etc. In the case of Mozart's map the silhouette of Mozart's head was chosen as the shape of the map. The map consists of 776 units, which are aligned according to this shape. For the clustering algorithm, a learning rate of 0.75 and an initial neighborhood radius of 20 were chosen, clustering was done in 25,000 iterations.

Description of the Map of Mozart

On the resulting map, pieces of music with similar features are mapped close to each other, and pieces with low similarity are located in distant regions. Groups of many songs with similar characteristics are building clusters. With the SDH visualization these clusters are exhibited as "islands" on the Map of Mozart. The further away two pieces are from each

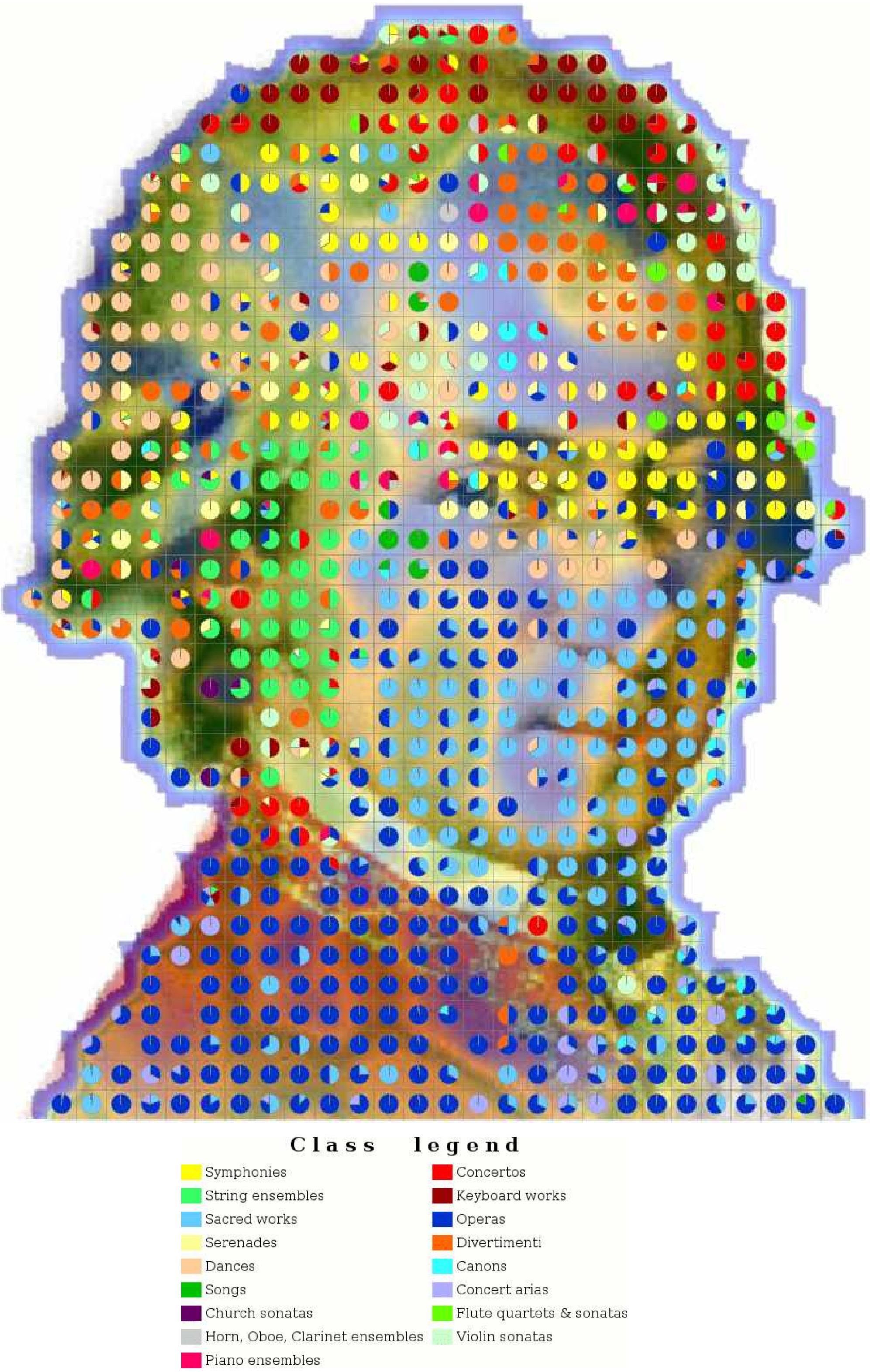


Figure 6.12: Map of Mozart (semi-transparent Smoothed Data Histograms (SDH) and background visualization, with categories as overlay)

other, the more divergent are their musical characteristics. If the pieces in the collection are not from clearly distinguishable categories the map reflects this by placing pieces along continuous transitions.

The map can be explored acoustically with the PlaySOM application (see Section 6.4), but with the use of the class overlay, it is possible to describe the structures of the Map of Mozart also by referring to an image. Figure 6.12 shows the Map of Mozart in a hybrid visualization, including Mozart's head, a semi-transparent SDH visualization and the class overlay.

In this image it can be seen clearly that almost all operas composed by Mozart are located in the lower part of the Map of Mozart. The operas are further divided into different regions, for example recitatives, located in the region of Mozart's neck. Operas with male voices are located at the left while female voices are clustered at the right. There is also a transition from the operas to sacred works.

One can find string ensembles in the region of Mozart's right ear, while the dances are arranged left-above of the string ensembles in the region of the back of the head. A cluster of piano music can be found on the top edge containing piano sonatas and piano concerts. Faster pieces are usually mapped more to the left than to the right, which is also the reason why symphonies are divided up into two clusters: *presto* (fast) pieces are located in a cluster at the top and *adagio* (slow) pieces in the area of Mozart's left ear (at the right).

It becomes apparent that the clustering abilities of the Self-Organizing Map and the features extracted by the Rhythm Pattern algorithm are working very well on an as homogeneous data set as this specific music collection by Mozart. The automatic organization of the complete works of Mozart is particularly remarkable as the algorithm was based solely on the audio content of the music, without considering any musical knowledge or meta-data.

The Mnemonic Map of Mozart offers attractive and eye-catching visualizations to the user, and a playful alternative to the Köchel-Verzeichnis for exploring Mozart's music.

In addition to the PlaySOM application which is used for exploration, an interactive web demo of the Map of Mozart has been created. In the web version the user can choose between different visualization variants: (1) the map with the image of Mozart as background, (2) only the shape of Mozart with the SDH visualization and (3) a combination of both, using a semi-transparent SDH on the image. Additionally, the user can choose to show the class distribution as an overlay to these visualizations. Even though the web version allows less ways of interaction, the user can still easily navigate through all the pieces of music on the map, and select music to listen to. The Map of Mozart can be explored with limited amounts of free music available online at `http://www.ifs.tuwien.ac.at/mir/mozart`.

6.7 Conclusions

In this chapter an approach for clustering music collections has been presented together with intuitive interactive applications based thereon.

The Self-Organizing Map has been introduced which is a clustering algorithm that maps high-dimensional data – which has been previously extracted from audio signals – to a two-dimensional map. A large range of visualizations is available to render the inherent structures and relations between songs in a music collection more explicitly in an intuitive manner.

Based on these forms of representing music libraries new applications have been created which allow for browsing music collections by similarity, without the need for meta-information such as artist names and song titles. Meta-data, of course, may be used in addition to the content-based approaches, if available, e.g. to search and locate particular music. Furthermore, a *Music Map* can be utilized directly to select and play back music. The model for the selection of paths through the “landscape” allows to create specific playlists for particular situations, e.g. a playlist for a relaxing dinner or for a party. This novel form of interaction with music libraries has also been ported to mobile devices, which enable the streaming of desired music from a repository to mobile players independently from one’s location.

My contributions to the software programs and applications were

- a query interface for the PlaySOM application that allows for meta-data based queries

- a modified Hit Histogram visualization that applies spline interpolation and is able to present results of different queries to the music collection in an attractive visualization similar to the SDH visualization

- tests and improvements concerning the usability of the PocketSOMPlayer on a PDA and a PDA phone
- the extraction of audio features from Mozart's complete works
- the training and creation of the MnemonicSOM-based *Map of Mozart* and its presentation on a web site
- porting of the *Map of Mozart* to the PocketSOMPlayer format

The applicability of the SOM-based approach to cluster music collections was demonstrated by the creation of the *"Map of Mozart"*, which drew attention from international press and media. The *Map of Mozart* clusters the complete works of W. A. Mozart on a specially shaped SOM (a Mnemonic SOM that takes the silhouette of Mozart's head) according to acoustical similarity which has been extracted from the music by an automatic audio feature extraction algorithm presented in Chapter 3 and evaluated in several benchmarks in Chapter 5. The automatic clustering of Mozart's music on the *Map of Mozart* shows the feasibility of the entire approach.

Chapter 7

Summary and Conclusions

In this thesis the problems of categorizing, organizing, searching in and interacting with music collections have been outlined. Chapter 1 illustrated the motivation for the work described in this thesis. In Chapter 2 publications related to this work have been reviewed, in particular approaches to feature extraction from audio, music classification, benchmarking, clustering and visualization of music archives as well as new interfaces to music collections. Various approaches to audio feature extraction have been explained in Chapter 3 including low-level temporal and spectral audio features, MPEG-7 audio descriptors, Mel Frequency Cepstral Coefficients, Wavelet Transform Features, Beat Histograms, Pitch Histograms, Statistical Spectrum Descriptors, Rhythm Patterns and Rhythm Histograms. The three latter feature sets have undergone detailed evaluations and benchmarkings described in Chapter 5. Chapter 4 provided a detailed overview of the music collections involved in benchmarking and evaluation, describing the distinct characteristics of those audio collections.

Chapter 5 started with a summarization of the efforts for establishing standard scientific benchmarkings in Music Information Retrieval research followed by a short introduction to classification approaches and appropriate measures. Starting with the Rhythm Patterns feature set, a number of benchmarkings have been performed. The baseline algorithm won the ISMIR 2004 Rhythm Classification contest. Subsequently, a study of the influence of psycho-acoustics in feature extraction has been conducted, with the result of identifying a number of psycho-acoustic transformations (in particular Decibel, Phon and Sone) as crucial for application in classification tasks, while discovering potential issues with others (e.g. Spectral Masking, smoothing). Two new feature sets have been presented: Statistical Spectrum Descriptors and Rhythm Histograms. These feature sets, alongside the improvements on the Rhythm Patterns feature set, have been evaluated thoroughly on three reference audio collections. The new feature sets have been characterized as being competitive in comparison to other state-of-the-art approaches. Multiple combinations of them were submitted to the MIREX 2005 evaluation on Genre Classification where it was competitive with several other approaches, outperformed only by the approaches of two other participants. In the MIREX 2006 benchmarking on Music Similarity Retrieval the Statistical Spectrum Descriptors were identified to perform equally well as the top 5 state-of-the-art algorithms, as determined by a statistical significance test on the results of a human listening test. Prior to MIREX 2006 a study has been performed on evaluating different distance metrics for similarity computations in music databases.

In Chapter 6 the devised, evaluated and benchmarked feature sets were applied to music organization applications, which utilize the Self-Organizing Map clustering algorithm. After introducing the Self-Organizing Map algorithm, a number of possible visualizations for exhibiting the clustered structures within music collections are described. The clustering is purely acoustics based, according to sound similarity inherent in the audio feature sets. The PlaySOM and PocketSOMPlayer applications were presented, which enable new forms of interaction with music collections. They support browsing, zooming and selection of clusters of similar music and consequently the discovery of new music based on music one already knows and one likes. Furthermore, the selection of trajectories through the clustered music collection facilitates the creation of playlists by music similarity, suitable for particular situations or moods.

For a demonstration of the practicability of both the clustering and feature extraction approaches, Mozart's complete works have been analyzed, clustered and organized on a Music Map, creating the *Map of Mozart*. Though this particular example contains music from one single composer and is thus supposed to be very homogeneous, the feature extraction and clustering approaches managed to organize the music very well, dividing the works of Mozart into different clusters containing e.g. operas, dances, symphonies, piano works, etc.

To summarize, the approaches presented for feature extraction from musical audio signals have been evaluated in established standard scientific benchmarkings and have proven to be very well suited for music classification tasks, for music similarity, as well as clustering-based organization of music on *Music Maps*. Consequently, they are valuable for novel music retrieval applications, which facilitate searching for music and relieve users from cumbersome manual annotation and categorization tasks.

List of Tables

List of Figures

Bibliography

[AAG06] Xavier Amatriain, Pau Arumí, and David Garcia. CLAM: A framework for efficient and rapid development of cross-platform audio applications. In *Proceedings of ACM Multimedia 2006*, Santa Barbara, CA, 2006.

[AD90] Paul Allen and Roger B. Dannenberg. Tracking musical beats in real time. In S. Arnold and G. Hair, editors, *Proceedings of the International Computer Music Conference (ICMC)*, pages 140–143, Glasgow, 1990.

[AHH$^+$01] Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, Thorsten Kastner, and Markus Cremer. Content-based identification of audio material using MPEG-7 low level description. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 197–204, Bloomington, IN, USA, October 15-17 2001.

[AP02] Jean-Julien Aucouturier and Francois Pachet. Music similarity measures: What's the use? In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, October 13-17 2002.

[Azi06] Taha Abdel Aziz. Coloring of the SOM based on class labels. Master's thesis, Vienna University of Technology, October 2006.

[BKLR06] Emmanouil Benetos, Constantine Kotropoulos, Thomas Lidy, and Andreas Rauber. Testing supervised classifiers based on non-negative matrix factorization to musical instrument classification. In *Proceeedings of the European Signal Processing Conference (EUSIPCO)*, Florence, Italy, September 4-8 2006.

[BLEW03] Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October 26-30 2003.

[BLEW04] Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[BSS04] Roberto Basili, Alfredo Serafini, and Armando Stellato. Classification of musical genre: a machine learning approach. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

[CFPT06] Matthew Cooper, Jonathan Foote, Elias Pampalk, and George Tzanetakis. Visualization in audio-based music information retrieval. *Computer Music Journal*, 30(2):42–62, 2006.

[CKGB02] Pedro Cano, Martin Kaltenbrunner, Fabien Gouyon, and Eloi Battle. On the use of fastmap for audio retrieval and browsing. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, October 13-17 2002.

[CMRR82] Chris Chafe, Bernard Mont-Reynaud, and Loren Rush. Toward an intelligent editor of digital audio: Recognition of musical constructs. *Computer Music Journal*, 6(1):30–41, 1982.

[CPL94] Piero Cosi, Giovanni De Poli, and Giampaola Lauzzana. Auditory modelling and self-organizing neural networks for timbre classification. *Journal of New Music Research*, 23(1):71–98, March 1994.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

[CW03] Holger Crysandt and Jens Wellhausen. Music classification with MPEG-7. In *Proceedings of SPIE-IS&T Electronic Imaging*, volume 5021 of *Storage and Retrieval for Media Databases*, pages 307–404, Santa Clara (CA), USA, January 2003. The International Society for Optical Engineering.

[DEH05] J. Stephen Downie, Andreas F. Ehmann, and Xiao Hu. Music-to-knowledge (M2K): a prototyping and evaluation environment for music digital library research. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, page 376, Denver, Colorado, USA, June 7-11 2005.

[DGW04] Simon Dixon, Fabien Gouyon, and Gerhard Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 509–516, Barcelona, Spain, October 2004.

[DH89] Peter Desain and Henkjan. Honing. The quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):56–66, 1989.

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[Dix99] Simon Dixon. A beat tracking system for audio signals. In *Proceedings of the Conference on Mathematical and Computational Methods in Music*, pages 101–110, Vienna, Austria, December 1999.

[Dix05] Simon Dixon. Live tracking of musical performances using on-line time warping. In *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx05)*, pages pp 92–97, Madrid, Spain, September 2005.

[DMR00] Michael Dittenbach, Dieter Merkl, and Andreas Rauber. The growing hierarchical self-organizing map. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume VI, pages 15 – 19, Como, Italy, July 24-27 2000.

[Dow02] J. Stephen Downie. Report on ISMIR 2002 conference panel I: Music information retrieval evaluation frameworks. *D-Lib Magazine*, 8(11), November 2002. ISSN 1082-9873.

[Dow03a] J. Stephen Downie. *Annual Review of Information Science and Technology*, volume 37, chapter Music information retrieval, pages 295–340. Information Today, Medford, NJ, 2003.

[Dow03b] J. Stephen Downie. Toward the scientific evaluation of music information retrieval systems. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October 26-30 2003.

[DPW03] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of dance music by periodicity patterns. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 159–165, Baltimore, Maryland, USA, October 26-30 2003.

[DTW97] Roger B. Dannenberg, Belinda Thom, and David Watson. A machine learning approach to musical style recognition. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 344–347, Thessaloniki, Greece, September 25-30 1997.

[EWBL02] Daniel P.W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, October 13-17 2002.

[FG94] Bernhard Feiten and Stefan Günzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18(3):53–65, 1994.

[Foo97] Jonathan T. Foote. Content-based retrieval of music and audio. In C.-C.J. Kuo, editor, *Proceedings of SPIE Multimedia Storage and Archiving Systems II*, volume 3229, pages 138–147, 1997.

[Fri37] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675?701, December 1937.

[FS95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the European Conference on Computational Learning Theory (EUROCOLT)*, pages 23–37, 1995.

[GD04] Fabien Gouyon and Simon Dixon. Dance music classification: A tempo-based approach. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

[GD05] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1), 2005.

[GDPW04] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, pages 196–204, London, UK, June 17-19 2004.

[GG05] Masataka Goto and Takayuki Goto. Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, September 11-15 2005.

[GHNO02] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, October 13-17 2002.

[GHNO03] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Music genre database and musical instrument sound database. In *Proceedings of* *the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October 26-30 2003.

[GKM03] E. Gomez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1), 2003.

[GM95] Masataka Goto and Yoichi Muraoka. A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference (ICMC)*, Banff, Canada, 1995.

[Gom06] Emilia Gomez. *Tonal Description of music audio signals.* PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[HMM$^+$05] H. Homburg, I. Mierswa, B. Moeller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, September 11-15 2005.

[Koh01] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.

[KPW04] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist classifiction with web-based data. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 10-14 2004.

[Lai06] David Laister. Optimierung der Merkmalsberechnung für Audio-Daten. Master's thesis, Vienna University of Technology, Vienna, Austria, October 2006.

[Lem95] Marc Leman. *Music and Schema Theory, Cognitive Foundations of Sytematic Musicology.* Number 31 in Springer Series in Information Science. Springer, Berlin, Heidelberg, 1995.

[LH78] Hugh Christopher Longuet-Higgins. The perception of music. *Interdisciplinary Science Reviews*, 3(2):148–156, 1978.

[LH00] Zhu Liu and Qian Huang. Content-based indexing and retrieval-by-example in audio. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, New York, USA, July 30 - Aug. 2 2000.

[LHL82] Hugh Christopher Longuet-Higgins and Christopher Lee. The perception of musical rhythms. *Perception*, 11:115–128, 1982.

[Lid03] Thomas Lidy. Marsyas and rhythm patterns: Evaluation of two music genre classification systems. In *Proceedings of the Fourth Workshop on Data Analysis (WDA2003)*, Ruzomberok, Slovak Republic, June 13-15 2003.

[Log00] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, Massachusetts, USA, October 23-25 2000.

[LOL03] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 282 – 289, Toronto, Canada, 2003.

[LPR05] Thomas Lidy, Georg Pölzlbauer, and Andreas Rauber. Sound re-synthesis from rhythm pattern features - audible insight into a music feature extraction process. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 93–96, Barcelona, Spain, September 5-9 2005.

[LR03] Arie Livshin and Xavier Rodet. The importance of cross database evaluation in musical instrument sound classification: A critical approach. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October 26-30 2003.

[LR05] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the Sixth International Conference on Music Information Retrieval*, pages 34–41, London, UK, September 11-15 2005.

[LS01] Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, August 2001.

[LT01] Chih-Chin Liu and Po-Jun Tsai. Content-based retrieval of mp3 music objects. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM 2001)*, pages 506 – 511, Atlanta, Georgia, 2001.

[LW01] Mingchun Liu and Chunru Wan. A study on content-based classification and retrieval of audio database. In *Proceedings of the 5th International Database Engineering and Applications Symposium (IDEAS 2001)*, Grenoble, France, 2001. IEEE.

[Mal99] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2nd edition, 1999.

[Mar04] José M. Martínez, editor. *MPEG-7 Overview (version 10). ISO/IEC JTC1/SC29/WG11N6828*. International Organisation for Standardisation, Palma de Mallorca, Spain, October 2004.

[MH91] Ray Meddis and Michael J. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. *Journal of the Acoustical Society of America*, 89(6):2866–2882, June 1991.

[Mit05] Dalibor Mitrovic. Discrimination and retrieval of environmental sounds. Master's thesis, Vienna University of Technology, Vienna, Austria, December 2005.

[MLR06] Rudolf Mayer, Thomas Lidy, and Andreas Rauber. The Map of Mozart. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Victoria, Canada, October 8-12 2006.

[MMB$^+$05] G. Martens, H. De Meyer, B. De Baets, M. Leman, M. Lesaffre, J-P. Martens, and T. De Mulder. Distance-based versus tree-based key recognition in musical audio. *Soft Computing*, 9(8):565–574, 2005.

[MMR05] Rudolf Mayer, Dieter Merkl, and Andreas Rauber. Mnemonic SOMs: Recognizable shapes for self-organizing maps. In *Proceedings of the Workshop On Self-Organizing Maps (WSOM)*, pages 131–138, Paris, France, September 5-8 2005.

[MoM06a] Alle Werke Mozarts auf einem Porträt. Kurier, April 11 2006.

[MoM06b] Mozart à la carte. GEO, June 2006.

[MoM06c] Musik liegt in der Landkarte. Der Standard, April 12 2006.

[MoM06d] Ordnung statt Chaos: Musik auf virtueller Landkarte. Spiegel online, April 9 2006. `http://www.spiegel.de/wissenschaft/mensch/0,1518,410324,00.html`.

[MoM06e] Und jetzt grüne Musik mit Zacken. Financial Times Deutschland, August 1 2006.

[Moo96] Todd K. Moon. The expectation-maximation algorithm. *IEEE Signal Processing Magazine*, pages 47–70, November 1996.

[MPE02] *Information technology - Multimedia content description interface - Part 4: Audio. ISO/IEC 15938-4:2002*. International Organisation for Standardisation, 2002.

[MUNS05] Fabian Mörchen, Alfred Ultsch, Mario Nöcker, and Christian Stamm. Databionic visualization of music collections according to perceptual distance. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, September 11-15 2005.

[NDR05] Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and PocketSOMPlayer – alternative interfaces to large music collections. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 618–623, London, UK, September 11-15 2005.

[NLR05] Robert Neumayer, Thomas Lidy, and Andreas Rauber. Content-based organization of digital audio collections. In *Proceedings of the 5th Open Workshop of MUSICNETWORK*, Vienna, Austria, July 4-5 2005.

[Ori06] Nicola Orio. *Music Retrieval: A Tutorial and Review*. Foundations and Trends in Information Retrieval. Now Publishers, September 2006.

[Pam01] Elias Pampalk. Islands of music: Analysis, organization, and visualization of music archives. Master's thesis, Vienna University of Technology, December 2001.

[Pam06] Elias Pampalk. *Computational Models of Music Similarity and their Application to Music Information Retrieval*. PhD thesis, Vienna University of Technology, Austria, March 2006.

[PDR06] Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. Advanced visualization of self-organizing maps with vector fields. *Neural Networks*, 19(6–7):911–922, July–August 2006.

[PDW03] Elias Pampalk, Simon Dixon, and Gerhard Widmer. On the evaluation of perceptual similarity measures for music. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-03)*, pages 7–12, London, UK, September 8-11 2003.

[PDW04] Elias Pampalk, Simon Dixon, and Gerhard Widmer. Exploring music collections by browsing different views. *Computer Music Journal*, 28(2):49–62, 2004.

[PFW05] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Hierarchical organization and description of music collections at the artist level. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, Vienna, Austria, September 18-23 2005.

[Poh05] Tim Pohle. Extraction of audio descriptors and their evaluation in music classification tasks. Master's thesis, Universität Kaiserslautern, 2005.

[PRM02] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based organization and visualization of music archives. In *Proceedings of ACM Multimedia 2002*, pages 570–579, Juan-les-Pins, France, December 1-6 2002.

[Pur05] Hendrik Purwins. *Profiles of Pitch Classes - Circularity of Relative Pitch and Key: Experiments, Models, Music Analysis, and Perspectives*. PhD thesis, Technische Universität Berlin, August 2005.

[RF01] Andreas Rauber and Markus Frühwirth. Automatically analyzing and organizing music archives. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, Darmstadt, Germany, September 4-8 2001. Springer.

[Ros58] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[RPM02] Andreas Rauber, Elias Pampalk, and Dieter Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 71–80, Paris, France, October 13-17 2002.

[RPM03] Andreas Rauber, Elias Pampalk, and Dieter Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, June 2003.

[RTG98] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[SAH79] M.R. Schröder, B.S. Atal, and J.L. Hall. Optimizing digital speech coders by exploiting masking properties of the human ear. *Journal of the Acoustical Society of America*, 66:1647–1652, 1979.

[Sch97] Eric D. Scheirer. Pulse tracking with a pitch tracker. In *Proceedings of the 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, NY, USA, October 1997.

[Sch98] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, January 1998.

[SP01] Christian Spevak and Richard Polfreman. Sound spotting - a frame-based approach. In *Proceedings of the Second International Symposium on Music Information Retrieval: ISMIR 2001*, pages 35–36, Bloomington, IN, USA, October 15-17 2001.

[SS97] Eric Scheirer and Malcolm Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 1331–1334, Munich, Germany, 1997.

[Ste77] Mark J. Steedman. The perception of musical rhythm and metre. *Perception*, 6:555–569, 1977.

[SZM06] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, 23(2):133–141, 2006.

[TC00a] George Tzanetakis and Perry Cook. 3D graphic tools for isolated sound collections. In *Proceedings of the Conference on Digital Audio Effects (DAFx)*, Verona, Italy, December 2000.

[TC00b] George Tzanetakis and Perry Cook. Sound analysis using MPEG compressed audio. In *Proceedings of the International Conference on Audio, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey, 2000.

[TEC01] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 205–210, Bloomington, Indiana, October 15-17 2001.

[TEC02a] George Tzanetakis, Andreye Ermolinskyi, and Perry Cook. Beyond the query-by-example paradigm: New query interfaces for music information retrieval. In *Proceedings of the International Computer Music Conference (ICMC)*, Gothenburg, Sweden, September 2002.

[TEC02b] George Tzanetakis, Andreye Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, October 13-17 2002.

[THA04] Marc Torrens, Patrick Hertzog, and Josep Lluis Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

[TK00] Tero Tolonen and Matti Karjalainen. A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716, November 2000.

[Tza02] George Tzanetakis. *Manipulation, Analysis and Retrieval Systems for Audio Signals.* PhD thesis, Computer Science Department, Princeton University, 2002.

[Ult99] Alfred Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps*, pages 33–46, 1999.

[Ult03a] Alfred Ultsch. Pareto density estimation: A density estimation for knowledge discovery. In Baier D. and Wernecke K.D., editors, *Innovations in Classification, Data Science,* *and Information Systems - Proceedings 27th Annual Conference of the German Classification Society (GfKL)*, pages 91–100, Berlin, Heidelberg, 2003.

[Ult03b] Alfred Ultsch. U*-matrix: a tool to visualize clusters in high dimensional data. Technical report, Departement of Mathematics and Computer Science, Philipps-University Marburg, 2003.

[US90] Alfred Ultsch and H. Peter Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Network Conference (INNC'90)*, pages 305–308, Dordrecht, Netherlands, 1990.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[vGVvdW04] Rob van Gulik, Fabio Vignoli, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

[WBKW96] Erlig Wold, Thom Blum, Douglas Keislar, and James Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, Fall 1996.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[XRDH03] Ziyou Xiong, Regunathan Radhakrishnan, Ajay Divakaran, and Thomas S. Huang. Comparing MFCC and MPEG-7 audio features for feature extraction, maximum likelihood HMM and entropic prior HMM for sports audio classification. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, 2003.

[ZF99] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics - Facts and Models*, volume 22 of *Springer Series of Information Sciences*. Springer, Berlin, 1999.