BTP THESIS

MUSIC INFORMATION RETRIEVAL (MIR)

A Dissertation presented at
Indian Institute of Technology in
Candidacy for the Degree of Bachelor of Technology

Under the supervision of

Dr. S.R.M Prasanna
Assistant Professor
Department of Electronics and Communication Engineering
Indian Institute of Technology
Guwahati, India

By

Abhinav Singh and Ashna Dhanda
Roll Numbers: 03010203 and 03010210

Bachelor of Technology Program
Electronics and Communication Department
Indian Institute of Technology
Guwahati, India

CERTIFICATE

It is certified that the work contained in the thesis entitled "**Music Information Retrieval (MIR)**", Submitted by **Abhinav Singh and Ashna Dhanda** with roll numbers **03010203** and **03010210**, in the partial fulfillment for the degree of Bachelor of Technology has been carried out under my supervision and this work has not been submitted elsewhere for any other degree.

Date:

Dr. S.R.M Prasanna

Assistant Professor

Electronics and Communication Department

Indian Institute of Technology

Guwahati, Assam

Acknowledgement

This work has been carried out during 2006-2007 at the Electronics and Communication Department of Indian Institute of Technology, Guwahati, Assam, India.

We wish to express our gratitude to Dr. S.R. Mahadeva Prasanna for making it possible for us to start working on the Music Information Retrieval problem in 2004, for his help and advice during this work, and for his contribution and moral support.

We wish to thank the staff of the Electronics and Communication Department of Indian Institute of Technology for their special help. Especially, we wish to thank Jharna Rani Rabha and Sanjib Das for setting an example for us both as researchers and as persons.

We wish to thank our parents for the unfailing emotional support they have provided us with, throughout the four years of engineering.

We would also like to thank all our friends and batch mates for their good company during our stay at IIT Guwahati. Especially, we wish to thank Akash Bhunchal, Mohit Jaju, G.Anil Kumar, Archit Arya whose moral support and good humour has made designing algorithms fun.

Our warmest thanks go to our dear friends E.Sowmya Sudha and Sashikanta Nayak for their support, love, and understanding during the intensive stages of putting this work together. And finally, we thank God Almighty, for whatever we have achieved in life is only through his help, and an expression of his will.

Abhinav Singh and Ashna Dhanda,
Electronics and Communication Department,
Indian Institute of Technology,
Guwahati,
Assam,
India.

CONTENTS

List of Figures

Abstract

The creation of huge databases coming from both restoration of existing analog archives and new content is demanding more and more reliable and fast tools for content analysis and description, to be used for searches, content queries and interactive access. In that context, musical genres are crucial descriptors since they have been widely used for years to organize music catalogues, libraries and music stores. Despite their use, musical genres remain a poorly defined concept, which make of the automatic classification problem a non-trivial task.

In first phase, we review the state-of-the-art in automatic genre classification and content description and present new concepts and algorithms under development for real world databases. As description of audio is a broad field that incorporates many techniques, an overview of the main directions in current research is given. However, a detailed study of automatic audio classification is conducted and a music genre classifier is designed. To evaluate the classifier, a general database is created comprising of about 2000 songs from 9 different musical genres.

The classification algorithm used for the development of music genre classifier is Vector Quantization, which is commonly used for the task. Based on feature's effectiveness, a robust musical genre classifier is designed and a classification accuracy of 86.77% is achieved over audio files from 9 different musical genres.

The availability of large music collections calls for ways to efficiently access and explore them. We present an approach which combines descriptors derived from audio analysis with meta-information to create different views of a collection. Such views can have a focus on timbre, rhythm, artist, style or other aspects of music. For each view the pieces of music are organized on a map in such a way that similar pieces are located close to each other. The maps are visualized using an Islands of Music metaphor where islands represent groups of similar pieces.

We demonstrate our approach on a small collection using a meta-information-based view and two views generated from audio analysis, namely, beat periodicity as an aspect of rhythm and spectral information as an aspect of timbre.

CHAPTER 1
Introduction

This chapter briefly describes the motivation and context of this thesis. In Section 1.1 the thesis is outlined including a summarization of the major contribution.

1.1 Motivation

The value of a large music collection is limited by how efficiently a user can explore it. Portable audio players can store over 20,000 songs and online music shops offer more than 1 million tracks. Furthermore, a number of new services are emerging which give users nearly unlimited access to music.

New tools are necessary to deal with this abundance of music. Of particular interest are tools which can give recommendations, create playlists, and organize music collections. One solution is to utilize the power of Internet communities with techniques such as collaborative filtering. Furthermore, communities can share playlists as well as lists of their favourite artists.

This thesis deals with tools that do not use input from the communities. The first part of the thesis is on computational models of Automatic Audio Genre Classification Systems. The second part of the thesis deals with the models of Audio-Based Music Similarity.

1.2 Music Information Retrieval (MIR)

Music Information Retrieval is an interdisciplinary research field which deals with techniques to search and retrieve music related data. A list of relevant topics can be found online. The topics include, among many others, computational models of music similarity and their applications.

The major MIR conference is the annual ISMIR International Conference on Music Information Retrieval which started in the year 2000. Many of the papers referenced in this thesis have been published at ISMIR conferences.

1.3 Music Related Services

The overall goal of this thesis is to support users in accessing and discovering music. There are a number of music services available which already provide this functionality. Particularly well known are Apple's iPod and the associated iTunes Music Store which offers its customers various ways of discovering music. Recently the billionth song was sold over the portal. Apple's iTunes, Amazon, and other online stores are changing the distribution channels for music. Brick and mortar stores cannot compete with unlimited shelf space, highly efficient recommendations based on customer profiles, and 24/7 opening hours (every day a year).

Furthermore, a number of services offer (nearly or completely free) access to huge music collections. Such services often have the form of easily personalizable Internet radio stations and are powerful tools to discover or simply enjoy new music. Such services include Yahoo! Launchcast or Musicmatch, Pandora and Last FM.

There are numerous other services which supply metadata (and automatically tag music), organize personal collections, create playlists, or support searching similar artists. These services, tools, or projects include FreeDB, MusicMoz, MusicBrainz, MusicMagic Mixer by Predixis, GraceNote's

recommendation engine, Musiclens, Liveplasma, and Gnoosic. Of particular interest are Internet platforms which allow communities to exchange their music experiences. Such platforms include UpTo11, LiveJournal, Audioscrobbler, Webjay, and MySpace.

All in all, tools which enhance the experience of listening to music are rapidly developing. In this context the automatic audio genre classification, content-based similarity measures and the applications described in this thesis are a small building block to improve these tools.

1.4 Outline of this Thesis

This thesis consists of two major chapters. Chapter 2 deals with automatic audio genre classification systems. First, an introduction to audio classification is given and state-of-the-art techniques are described. Second, combinations of these techniques have been used and are optimized and evaluated. The presented evaluation consists of more than 2000 pieces from 9 genres. The appropriateness of using genre classification based evaluations to explore large parameter spaces is confirmed by a listening test. In particular, the results of the listening test show that the differences measured through genre-based evaluations correspond to human listeners' ratings.

Chapter 3 deals with automatic music organization systems. While in chapter 2, concentration was on audio classification based on genres, in chapter 3 we have presented an approach through which we can organize our music databases according to mood and styles. First, an introduction to music similarity and related work is given and state-of-art algorithms are described. Second, we combine some of these techniques to build a interactive user interface for organization of music databases.

In Chapter 4 we conclude the thesis work. In the first part we draw conclusions from work done in Chapter 2. Further we conclude chapter 4, with summary and outcomes of Chapter 3. In the end we discuss a few applications of our work done in real life.

Finally in Chapter 5 we discuss the references referred during this thesis work.

CHAPTER 2
Automatic Audio Genre Classification Systems

The creation of huge databases coming from both restoration of existing analog archives and new content is demanding more and more reliable and fast tools for content analysis and description, to be used for searches, content queries and interactive access. In that context, musical genres are crucial descriptors since they have been widely used for years to organize music catalogues, libraries and music stores. Despite their use, musical genres remain a poorly defined concept, which make of the automatic classification problem a non-trivial task.

In this chapter, we review the state-of-the-art in automatic genre classification and content description and present new concepts and algorithms under development for real world databases. As description of audio is a broad field that incorporates many techniques, an overview of the main directions in current research is given. However, a detailed study of automatic audio classification is conducted and a music genre classifier is designed. To evaluate the classifier, a general database is created comprising of about 2000 songs from 9 different musical genres.

The classification algorithm used for the development of music genre classifier is Vector Quantization, which is commonly used for the task. Based on feature's effectiveness, a robust musical genre classifier is designed and a classification accuracy of 86.77% is achieved over audio files from 9 different musical genres.

2.1 Introduction

Our daily life is highly dependent on information, for example in formats as text and multimedia. We need information for common routines as watching/reading the news, listening to radio, watching a video etc. However, we easily run into problems when a certain type of information is needed. The immense flow of information makes it hard to find what you are looking for.

2.1.1 Background

The rapid increase of information imposes new demands of content management as the media archives and consumer products begin to be very complex and hard to handle. Currently people perform searches in databases with different meta-tags, which only describe whole chunk of information with a constellation of tags. The meta-tags are constructed by different individuals and one realizes that the interpretation of meta-tags can differ from individual to individual. An automatic system that describes audio would systematize labeling and would also allow searches on the actual data, not just on labels of it. Better content management is the goal of the automatic audio description systems. Some commercial content management applications are already out but many possible applications are still undeveloped. *For example*, a person is listening to the radio and wants to listen to jazz. Unfortunately, all the radio stations play pop music mixed with advertisements. The listener gives up searching for jazz and gets stuck with the pop music. This problem can be solved with an automatic audio description system. The scenario may then change to following. The person that wants to listen to jazz only finds pop music on all the tuned radio stations. The listener then presses a 'search for jazz' button on the receiver and after a couple of seconds the receiver change radio station and jazz flows out of speaker. This examples show how content management may be an efficient tool that simplifies daily routines involving information management.

Musical genres are categories that have arisen through a complex interplay of cultures, artists and market forces to characterize similarities between musicians or compositions and organize music collections. Yet, the boundaries between genres still remain fuzzy as well as their definition making the problem of automatic classification a non-trivial task. The Music Genre Classification problem asks for taxonomy of genres i.e. a hierarchical set of categories to be mapped onto a music collection. Pachet and Cazaly [2] studied a number of musical genre taxonomies used in industry and on the Internet and showed that it is not straightforward to build up such a hierarchy of genres. As a good classification relies on a carefully thought taxonomy, we start here from a discussion on a number of critical issues.

Music Genre can be classified by two ways: *Non - Musical Criteria* and *Musical criteria*. Music may also be categorized by non-musical criteria such as geographical origin though a single geographical category will normally include a wide variety of sub-genres. It can also be said that a music genre (or subgenre) is defined by the techniques, the styles, the context and the themes (content, spirit). We further discuss some of the important aspects associated with some musical genres around the world.

- *Classical Music (European Classical Music)*

In common usage classical music often refers to orchestral music in general, regardless of when it was composed or for what purpose. A full size orchestra (about 104 players) may sometimes be called a "symphony orchestra". The typical orchestra consists of four proportionate groups of similar musical instruments, generally appearing in the musical score in the following order:
Woodwinds: 2 flutes, 2 clarinets, bass clarinet, etc
Brass: 5 trumpets, 2 to 6 horns etc.
Percussion: Snare drum, bass drum, timpani etc.
Strings: violins, harps, cellos, double basses, pianos etc.
Flutes, clarinets, horns, trumpets, timpani, violins, cellos are among the core symphonic instruments.

- *Jazz Music*

Jazz is a musical form that grew out of cross-fertilization of folk, blues and other music. Jazz is primarily an instrumental form of music. The instrument most closely associated with jazz may be the saxophone, followed by the trumpets. The piano, guitar and drums are also primary jazz instruments. The single most distinguishing characteristics of jazz are improvisation. Jazz also tend to utilize complex chord structures and an advanced sense of harmony.

- *Rock Music*

Rock in its broadest sense can refer to almost all popular music recorded since 1950. Its main feature includes an emphasis on rhythm and the use of the electric guitars. Rock is a form of popular music from the late 20th century which typically features a vocal melody (often with vocal harmony) that is supported by accompaniment of electric guitars, a bass guitar, and drums, often with a strong back beat. Keyboard instruments such as organ, piano, or synthesizers are often used in many types of rock music. While brass instruments, such as saxophone were common in some styles in earlier development of rock, they are less common in the newer subgenres of rock music since the 1990s. The genre of rock music is broad, and its boundaries loosely-defined, with related genres such as soul and funk sometimes being included in the definition of the term.

- *Electronic Music*

Electronic music is a term for music created using electronic devices. As defined by the IEEE standards body, electronic devices are low-power systems and use components such as transistors and integrated circuits. Working from this definition, distinction can be made between instruments that produce sound through electromechanical means as opposed to instruments that produce sound using electronic components. Examples of an electromechanical instrument are the teleharmonium,

Hammond B3, and the electric guitar, whereas examples of an electronic instrument are a Theremin, synthesizer, and a computer.

Pachet and Cazaly [2] showed that a general agreement on genre taxonomies does not exist. Taking the example of well known websites like AllMusic (531 Genres), Amazon (719 Genres), and Mp3 (430 Genres), they only found 70 terms common to the 3 taxonomies. Furthermore, genre taxonomies may be dependant on cultural references. For example, a song by the French singer Charles Aznavour would be considered as "Variety" in France but would be filed as "World Music" in the UK. Due to difficulty of defining a universal taxonomy, more reasonable goals must be considered. In fact, Pachet and Cazaly eventually gave up their initial goal to define a general taxonomy of musical genres [2] and Pachet and al. decided to use simple two-level genre taxonomy of 20 genres and 250 subgenres in the context of the Cuidado music browser [3].

Musical genres are the main top-level descriptors used by music dealers and librarians to organize their music collections. Though they may represent a simplification of one artist's musical discourse, they are of a great interest as summaries of some shared characteristics in music pieces. With Electronic Music Distribution (EMD), music catalogues tend to become huge (the biggest online services propose around 1 million tracks); in that context, associating a genre to a musical piece is crucial to help users finding what they are looking for. In fact, the amount of digital music urges for efficient ways to browse, organize and dynamically update collections: it definitely requires new means for automatic annotation. In the case of music genre annotation, Weare [1] reports that the manually labeling of hundred thousand songs for Microsoft's MSN Music Search Engine needed about 30 musicologists for one year. At the same time, even if terms such as *jazz*, *rock* or *pop* are widely used, they often remain loosely defined so that the problem of automatic genre classification becomes a non-trivial task

2.1.2 Related Work

2.1.2.1 Feature Extraction

In the digital media world, generic audio information is mostly represented by bits allowing a direct reconstruction of an analogue waveform. In real world applications a precise symbolic representation of a (new) song is rarely available and one has to deal directly with most straightforward form, i.e. audio samples. Audio samples, though sampling the exact sound waveform, can not be used directly by automatic analysis systems because of the low level and low "density" of the information they contain; put in another way, the amount of data is huge and the information contained in audio samples taken independently is too small to deal with humans at the perceptual layer.

The first step of analysis systems is thus to extract some *features* from the audio data to manipulate more meaningful information and to reduce the further processing. Extracting features is the first step of most pattern recognition systems. Indeed, once significant features are extracted, any classification scheme may be used. In the case of audio signals, features may be related to the main elements of music including *melody, harmony, rhythm, timbre and spatial location.*

2.1.2.1.1 Timbral Features

Timbre is currently defined in literature as the perceptual feature that makes two sounds with the same pitch and loudness sound different. Features characterizing timbre analyze the spectral distribution of the signal though some of them are computed in the time domain. These features are global in the sense that they integrate the information of all sources and instruments at the same time. Most of these descriptors are computed at regular time intervals, over short windows of typical length between 10 and 30 ms.

- *Zero-Crossing Rate [4], [6]:* it is defined as the number of zero-crossings of the signal in the time domain; it is a measure of noisiness of the signal and it is correlated with pitch. Algorithm is simple and has low computational complexity. Scheirer use the ZCR to classify audio between speech/music, Tzanetakis [17] use it to classify audio into different genres of music and Gouyon use it to classify percussive sounds. Some part of music has variations in ZCR. For instance, a drum intro is a pop song can have high variation in ZCR values.
- *RMS [6]:* It is a measurement of the energy of a signal. The RMS value is however defined to be the square root of the average of a squared signal.
- *Loudness [6]:* simple models of loudness consist typically of an exponentiation of the energy of the frame.
- *Low Energy Feature [4], [6], [15]:* it is the percentage of frames within a larger window that have RMS energy lower than the mean RMS energy across the window.
- *Linear Prediction Coefficients [7], [8*]: linear prediction has been studied in the context of speech recognition to model sound production: the observed sound is supposed to be the result of the linear filtering of a simple signal. The estimated coefficients of the filter can be used as timbre descriptors since they encode the effect of the resonating body of the instrument (or of the vocal track in the case of speech production).
- *FFT coefficients [8]:* the feature vector is simply the vector of the FFT coefficients.
- *Mel-Frequency Cepstrum Coefficients (MFCCs) [4], [5], [6], [7], [9], [14], [15]:* they are perceptually motivated features obtained by taking the log-amplitude of the magnitude spectrum, warping the spectrum onto a perceptual frequency scale (the Mel frequency scale) and by applying a discrete cosine transform on the Mel coefficients to decorrelate the resulting feature vector. Thirteen coefficients are typically used for speech recognition.
- *Spectral Centroid [4], [6]:* it describes the center of gravity of the power spectrum. It is a cheap description of the shape of the power spectrum. It indicates whether an audio spectrum is dominated by low or high frequencies and additionally it is correlated with a major perceptual dimension of timbre; i.e. sharpness.
- *Spectral Roll-Off [4], [6]:* it is a measure of spectral shape. It is defined as the frequency below which most of the power spectrum is concentrated (typically 85 % of the power spectrum).
- *Spectral Flux [4], [6*]: it is a measure of the amount of local spectral change. It is evaluated as the difference between the normalized magnitudes of successive spectral distributions.
- *Spectrum Spread [6]:* it describes the second moment of the power spectrum. It is a cheap descriptor of the shape of the power spectrum that indicates whether it is concentrated in the vicinity of its centroid, or spread out over the spectrum. It allows differentiating between tone-like and noise-like sounds.
- *Spectrum Flatness [6]:* it expresses the deviation of the signal's power spectrum over frequency from a flat shape (corresponding to a noise-like or an impulse-like signal). A high deviation from a flat shape may indicate the presence of tonal components.
- *Harmonic Ratio [6]:* it is loosely defined as the proportion of harmonic components within the power spectrum. It is derived from the correlation between the signal and a lagged representation of the signal, lagged by the fundamental period of the signal.
- *Wavelet [10]:* the wavelet decomposition scheme allows a subdivision of the signal into octave sub-bands while providing good time and frequency resolution. The Daubechies Wavelet Coefficients Histograms (DWCH's) proposed in [10] are histograms of wavelet coefficients over a large window from which features are extracted by evaluating moments.

Most of the proposed algorithms for musical genre classification use indeed one small segment of audio per title: typically a 30-second long segment starting 30 seconds after the beginning of the piece to avoid introductions that may not be representative of the whole piece. Extraction of high-level descriptors from unrestricted polyphonic audio signals is not yet state of the art. Thus most

approaches focus on timbre modeling based on combinations of low-level descriptors. Timbre may contain sufficient information to roughly characterize musical genres as research demonstrated that humans with little to moderate musical training were able to perform a correct classification of music (among 10 genres) in 53 % of the cases after listening to only 250 milliseconds and in 72% of cases based on only 3 seconds of audio [12]. This suggests that no high-level understanding of music is needed to characterize genres as 250 milliseconds and in a lesser manner 3 seconds are too little time to recognize a musical structure.

2.1.2.2 Pattern Classifier

Extracting features is the first step of most pattern recognition systems. Indeed, once significant features are extracted, any classification scheme may be used. In the case of audio signals, a number of classifiers have been used in the literature. We further describe a few of them in this report:

2.1.2.2.1 The Unsupervised Algorithms

In the last few years, the machine learning approach has garnered increasing interest. In the unsupervised approach, an audio title is represented by a set of features as seen in section 1.2.1 and a similarity measure is used to compare titles among each others. Unsupervised clustering algorithms take then advantage of the similarity measure to organize the music collection with clusters of similar titles.

Similarity Measures

The simplest choice to measure distance between two feature vectors is to use a Euclidean distance or a cosine distance for example. However these distances will only make sense if the feature vectors are time-invariant. Otherwise two perceptually similar titles may be distant according to the measure if the similar features are time-shifted. A possible solution to build a time-invariant representation of a time series of feature vectors is to firstly build statistical models of the distribution of the features and then use the distance to compare these models directly. Typical models include K-means, Gaussian and Gaussian mixtures (GMMs).

Clustering Algorithms

K-means is probably the simplest and most popular clustering algorithm. It allows partitioning a set of vectors into K disjoint subsets. One of its weaknesses is that it requires the number of clusters (K) to be known in advance. Shao et al. [7] cluster their music collection with the Agglomerative Hierarchical Clustering, a clustering algorithm that starts with N singleton clusters (where N is the number of titles of the database) and that forms a sequence of clusters by successive merging. The Self-Organizing Map (SOM) and the Growing Hierarchical Self-Organizing Map (GHSOM) are used to cluster data and organize them on a 2-dimensional space in such a way that similar feature vectors are grouped close together. SOMs are unsupervised artificial neural networks that map high dimensional input data onto lower-dimensional output spaces while preserving the topological relationships between the input data items as faithfully as possible.

In some terms, the major drawback of unsupervised techniques can be that the obtained clusters are not labelled. In any case, the obtained clusters do not always reflect genre hierarchies, rather similarities dependent on the type of features (rhythmical similarities, melodic similarities, etc.).

2.1.2.2.2 The Supervised Algorithms

The supervised approach to music genre classification has been studied more extensively. The methods of this group suppose that taxonomy of genres is given and they try to map a database of songs into it by machine learning algorithms. As a first step, the system is trained with some manually labeled data, and then it is used to classify unlabelled data.

We describe here a number of commonly used supervised machine learning algorithms. We do not pretend to make an exhaustive list of such algorithms but to focus on those that have been used in the context of music genre classification.

Supervised Classifiers

- *K-Nearest Neighbor (KNN):* it is a non-parametric classifier based on the idea that a small number of neighbours influence the decision on a point. More precisely, for a given feature vector in the target set, the K closest vectors in the training set are selected (according to some distance measures) and the target feature vector is assigned the label of the most represented class in the K neighbours. KNNs are evaluated in the context of genre classification in [13], [10], [4].
- *Gaussian Mixture Models (GMM):* GMMs model the distribution of feature vectors. For each class, the existence is assumed of a probability density function expressible as a mixture of a number of multi-dimensional Gaussian distributions. The iterative Expectation Maximization (EM) algorithm is used to estimate the parameters for each Gaussian component and the mixture weights. GMMs have been widely used in the music information retrieval community, notably to build timbre models as seen in above section. They have been used to model directly musical genres in [10], [4], [5]. In [6], a tree-like structure of GMMs is used to model the underlying genre taxonomy: a divide-and-conquer strategy is used to first classify items on a coarse level and then on successively finer levels. The classification decision is thus decomposed into a number of local routing or refinement decisions in the taxonomy. In addition, feature selection at every refinement level allows optimizing classification results
- *Support Vector Machines (SVM*): SVMs are based on two properties: margin maximization (which allows for a good generalization of the classifier) and nonlinear transformation of the feature space with kernels (as a data set is more easily separable in a high dimensional feature space). SVMs have been notably used in the context of genre classification by [10], [14], [15].
- *Hidden Markov Model (HMM):* Hidden Markov Models can also be used for classification purposes. They have been extensively used in speech recognition because of their capacity to handle time series data. HMMs may be seen as a double embedded stochastic process: one process is not directly observable (hidden) and can only be observed through another stochastic process (observable) that produces the time set of observations. Though they may be well suited to modeling music, to our knowledge, HMMs have only been used in [9] and [14] for genre classification of audio content (they have been used in [7] as well but in the case of unsupervised organization of a music collection).
- *Mixture of Experts (ME):* A Mixture of Experts solves a classification problem by using a number of classifiers to decompose it into a series of sub-problems. Not only does it reduce the complexity of each single task but it also improves the global accuracy by combining the results of the different classifiers (experts). Of course, the number of needed classifiers is increased but having each of them a simpler problem to handle; the overall required computational power is reduced. Using a mixture of classifiers, each subtask may focus either on a subset of the attributes (feature selection), on different sample data (resampling, i.e. sub-sampling, bagging, boosting…), or on a different data labelling (decomposition of polychotomies into dichotomies). A range of solutions has been proposed in literature for the combination of different models into a global system. A possible solution is to use a majority vote of the different experts. This solution is used in [13] where 3 different MLPs characterize three different segments of a single song.

2.2 Our Automatic Audio Genre Classification Systems

In this section we present the algorithms used to develop a robust Automatic Audio Genre Classifier for associating automatically a music genre to an audio excerpt. Our algorithm parameterizes audio content by extracting 2 set of features describing 2 different dimensions of music: timbre and energy. Once features extracted, Vector Quantization is used for classification into musical genres. The underlying idea is to use separate models to approximate different parts of a problem and to combine the outputs from the models finally.

This section is organized as follows: In 2.1 we discuss the feature extraction techniques used for extracting features from the audio excerpt. Then in section 2.2 we discuss the classifier used for classification of the feature sets.

2.2.1 Feature Extraction

2.2.1.1 Segmentation into Analysis Frames

The audio excerpts used are sampled at 44100 Hz, 16-bit resolution and converted to mono signals. The first 30 seconds of the signals are discarded to avoid introduction that may not be representative of the rest of the excerpt. Only the next 30 seconds of the signal are kept for further analysis to limit further processing. The resulting signals are then analyzed through sliding windows of 23 ms overlapped by 50%. In the case of genre classification, it is probable that these precision requirements could be relaxed. West and Cox [16] use audio signals sampled at 22050 Hz and no overlap between the frames without significant loss of the classification accuracy. Further experiments have to be run in our case to check if the system is robust to signals with reduced quality.

2.2.1.2 Texture Windows

Frames of 23 ms are used for short time Fourier transforms analysis since they all representing the evolution of spectrum with a good precision. Yet this time scale too many variations occur. Some integration process must be held to build more robust features. Not only does it reduce further computations but it is also more perceptually relevant. Consequently, *texture windows* are used to combine low-level features of adjacent analysis frames.

The impact of the size of the window over classification accuracy has been studied in [17]. The conclusion is that texture windows of 1 second are a good compromise since no significant gain in classification accuracy is obtained by taking larger windows while the accuracy decreases as the window is shortened.

Scaringella[18] made experiments with the texture windows centred on the time positions of musical beats. The sizes of the corresponding windows were selected in accordance with the local beat rate of the excerpt. Though this may allow a perceptually more relevant modeling of musical signals, no significant improvement of classification accuracy has been obtained with this technique, probably because of the weakness of the state-of-the-art beat tracker. Consequently, the algorithm presented here use simple 1-second texture windows.

2.2.1.3 Timbre Features

Mel-Frequency Cepstral Coefficients (MFCC's) are computed from the analysis frames. MFCC's are widely used descriptor for timbre modeling coming from the speech recognition literature [19]. Each analysis frame is parameterized with 20 MFCC's. The number of MFCC's used has been chosen to limit the further computations rather than by a careful analysis of its impact on the classification accuracy, though the number of MFCC's is a subject of debate in the literature [20]. Mean, Standard

deviation over the texture window are evaluated for each MFCC resulting in 3 different feature vector sets each of 20 dimensions.

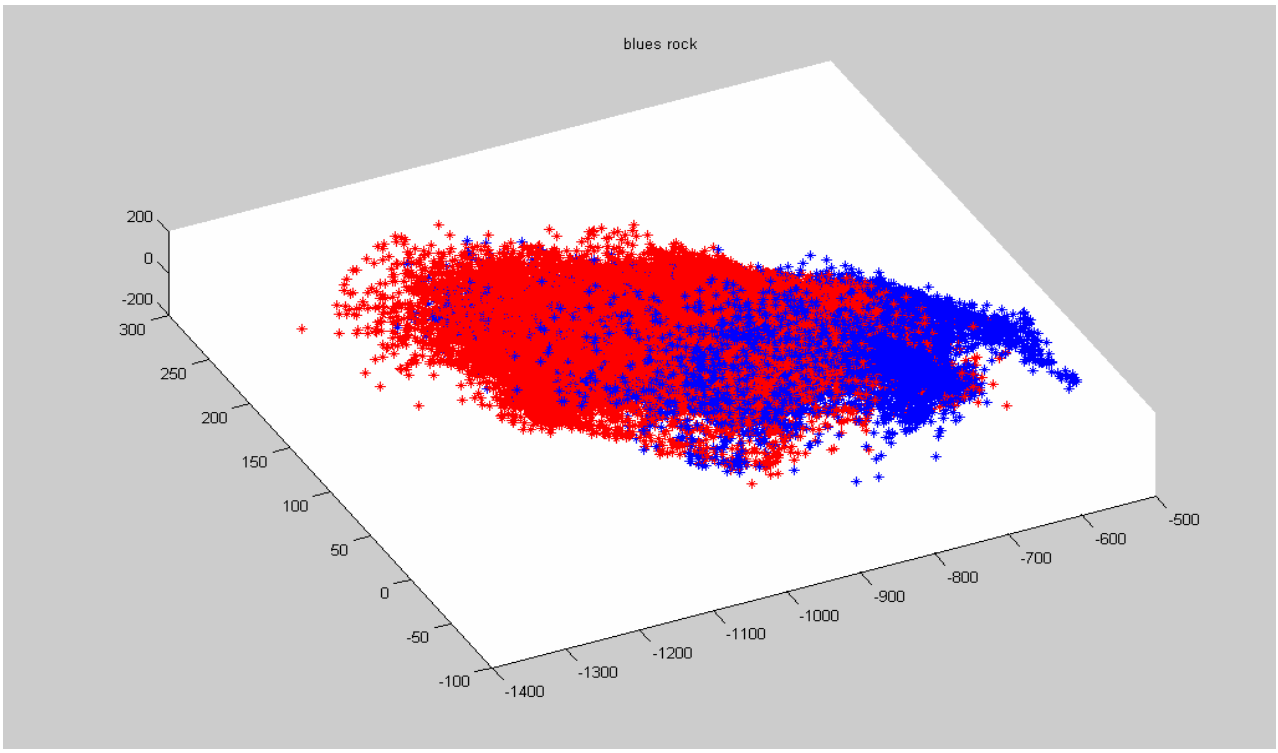

Figure 1. Blues v/s Rock

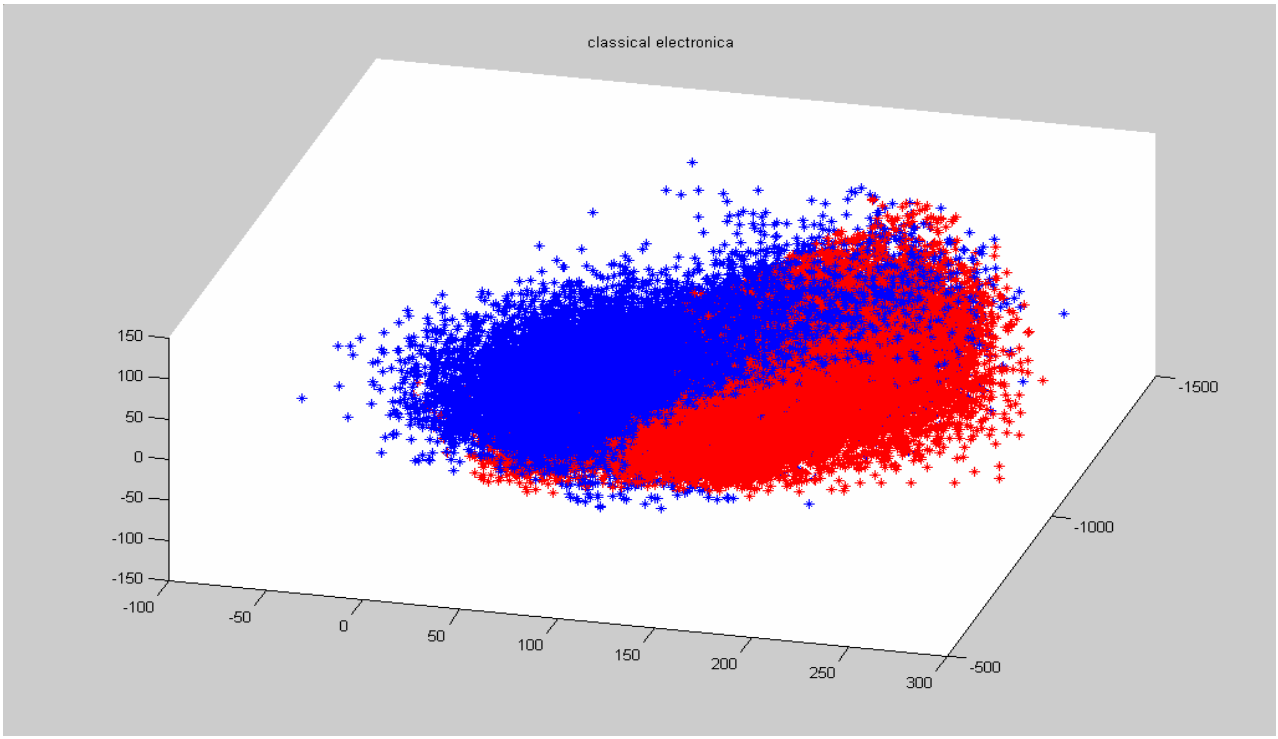

Figure 2. Classical v/s Electronic

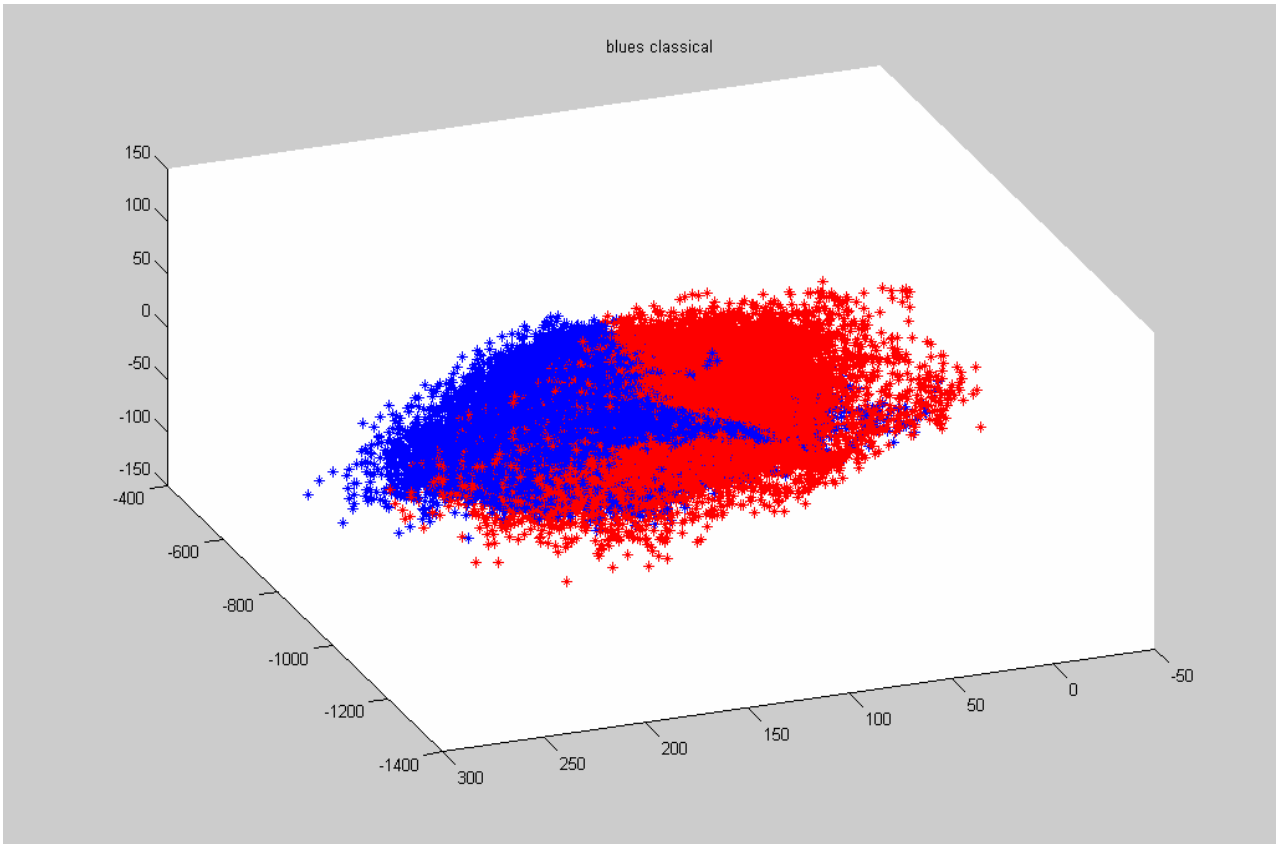

Figure 3. Blues v/s Classical

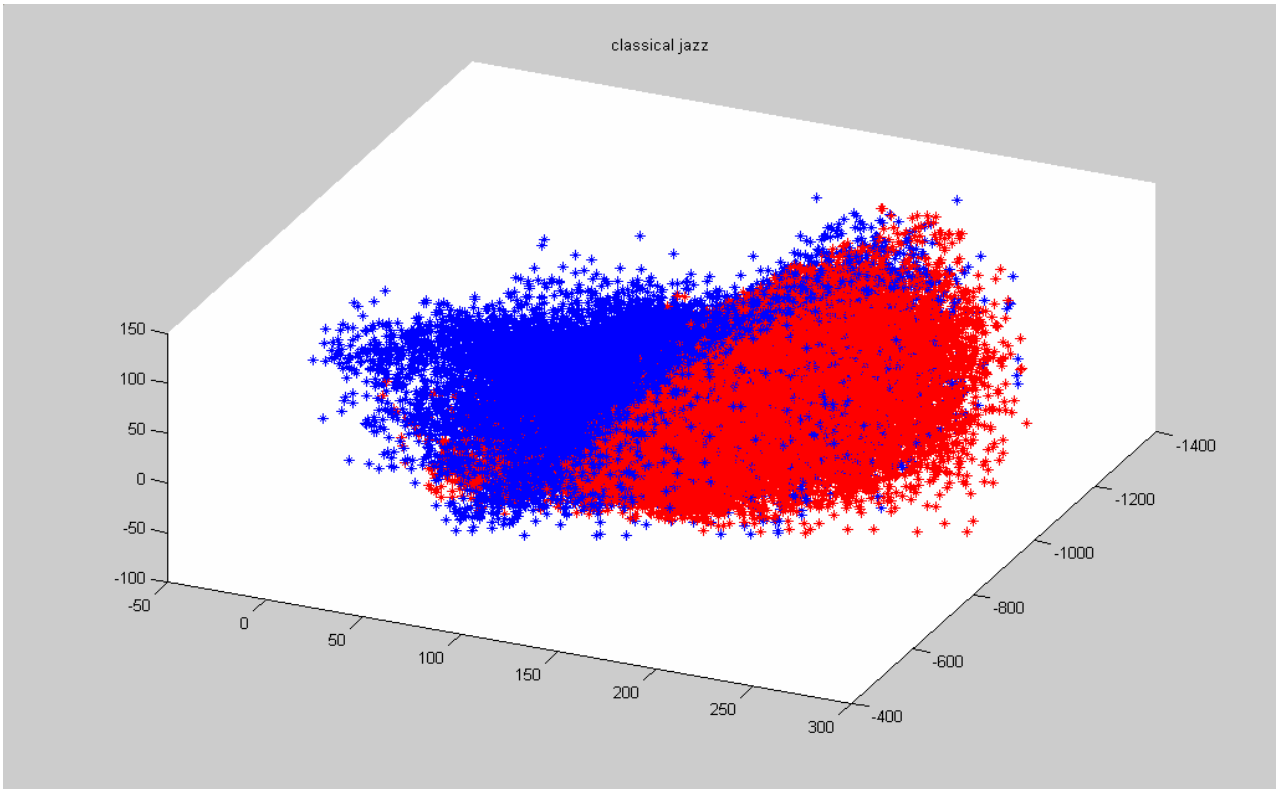

Figure 4. Classical v/s Jazz

2.2.1.4 Energy Features

As we musician play music in a particular octave for a given song, we tried out extracting energy in 6 octaves that span from 62.5 Hz to 22050 Hz. Log-compressed energies in 6 frequency bands are extracted from each analysis frame. Each band covers roughly one octave. Mean. Standard deviations of each coefficient are evaluated over the texture window.

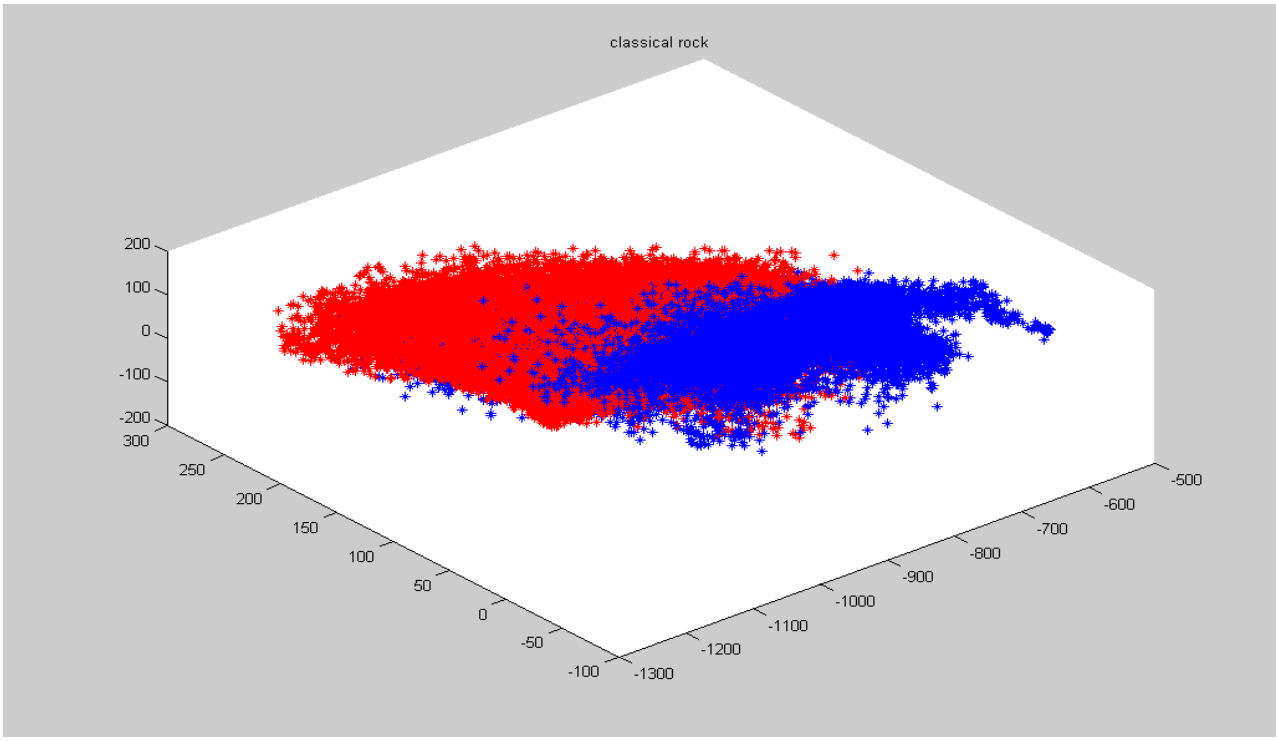

Figure 5. Classical v/s Rock

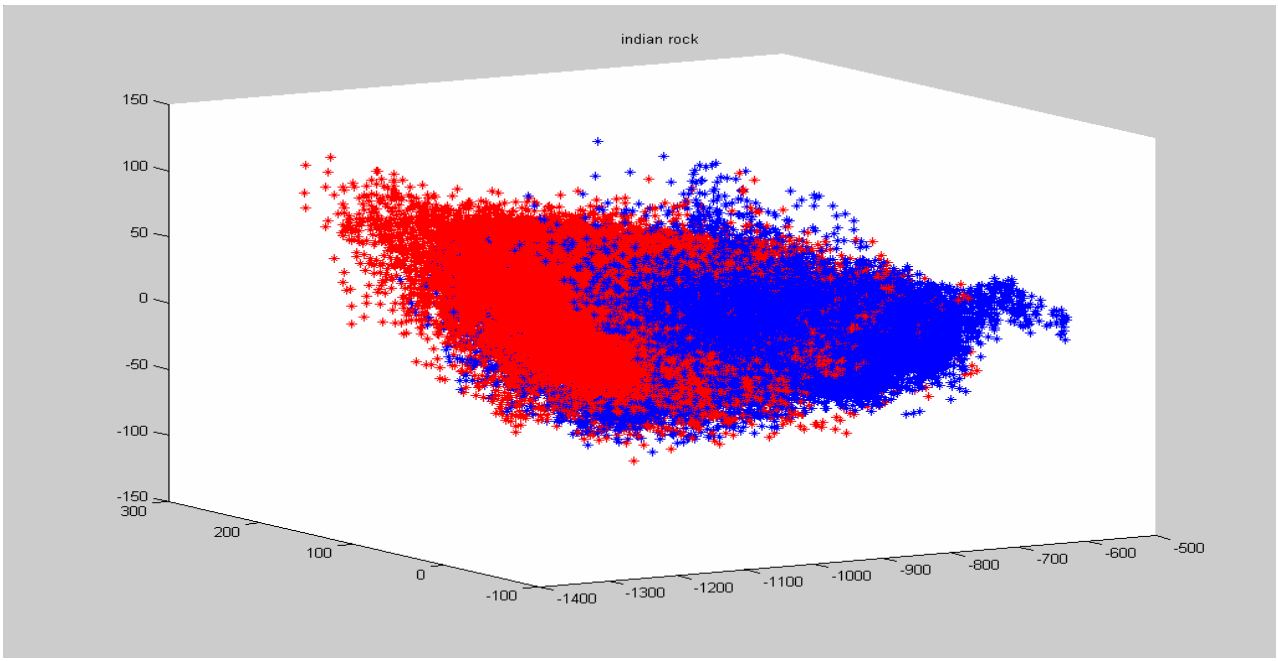

Figure 6. Ambient v/s Rock

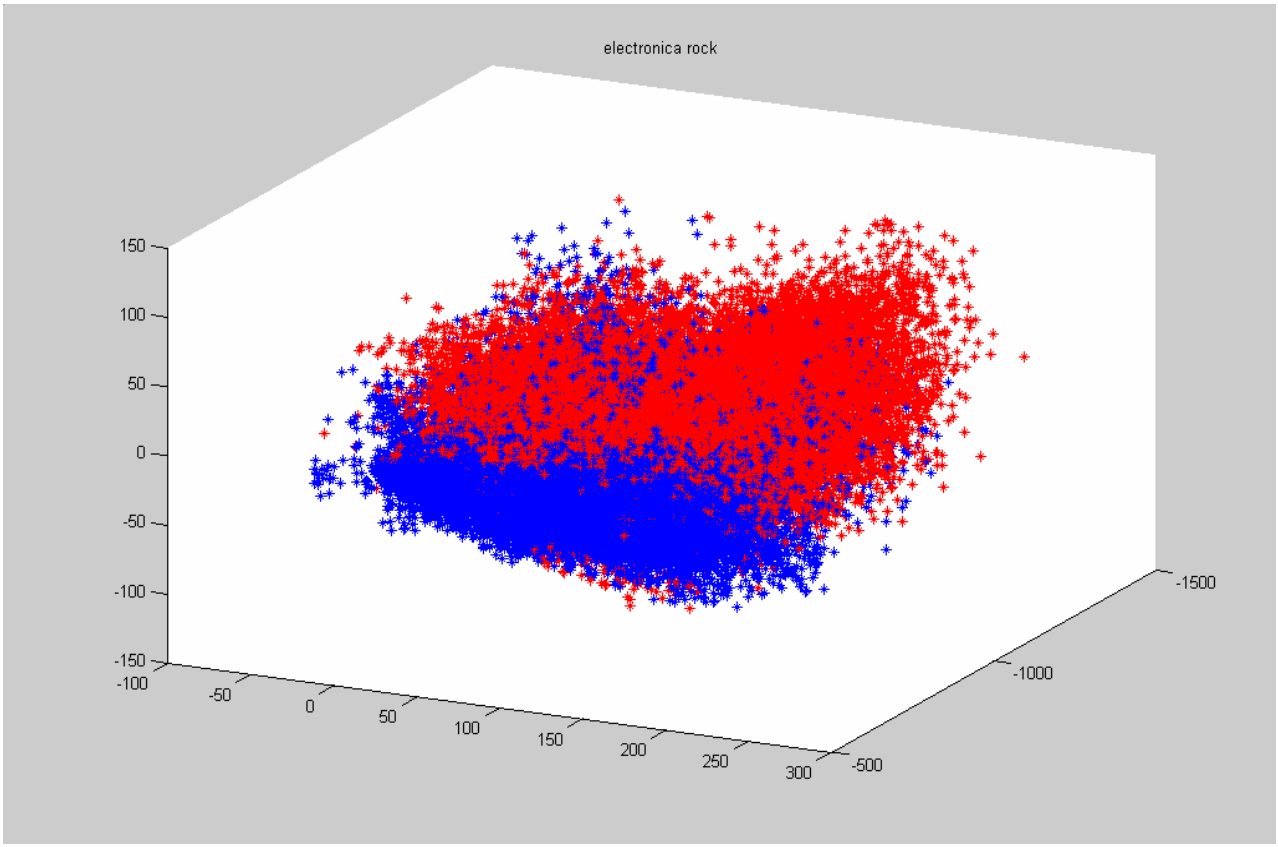

Figure 7. Electronic v/s Rock

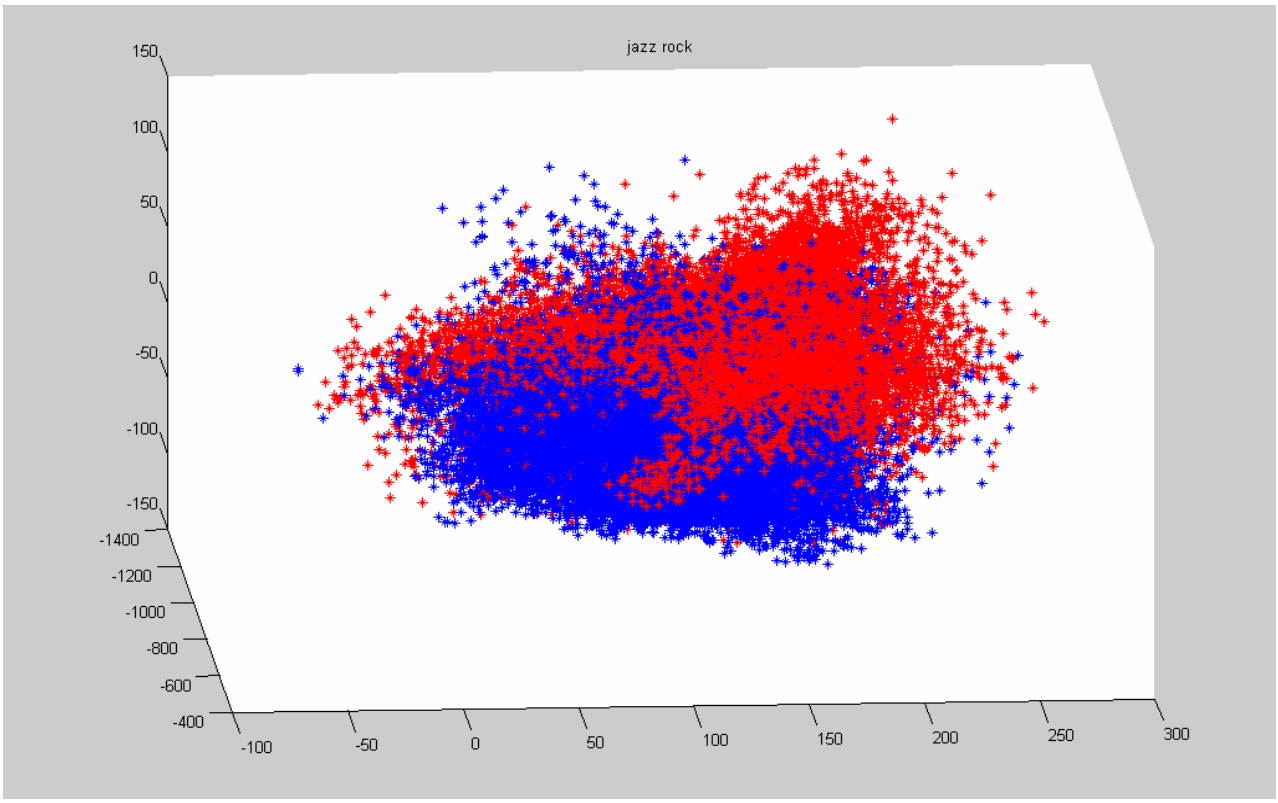

Figure 8. Jazz v/s Rock

2.2.2 Classification

Second step in building a robust classification system is to input the feature values extracted above into a classifier which operates a set of rules to generate various codebooks. In this project we have used Vector Quantization (VQ) as a classifier, which is the most widely used classification technique in the field of speech and audio recognition/classifier. Vector quantization is basically a concatenation of Binary Split and K-Means algorithms.

From feature extraction step, we extract two sets of feature vectors for each genre respectively. These feature vectors are saved in files named ***genre.mfcc*** and ***genre.energy*** for all the 9 genres. The dimension of these feature vector files are generally of the order 43000 x 20 for *genre.mfcc* and 43000 x 7 for *genre.energy*. It is not recommended computationally to compare an input test feature vector with all 43000 feature vectors. Hence we use Vector Quantization to extract 128 codebook vectors corresponding to each feature vector file. Finally we have two codebooks corresponding to each genre. These codebook vectors are saved in two files named ***genre.mfcc.codebook*** and ***genre.energy.codebook.*** The size of these codebooks is 128 x 20 for genre.mfcc.codebook and 128 x 7 for genre.energy.codebook.

In the testing phase, we extract the same sets of feature vector for the input audio files. For a 30 second test audio file, we generally get about 2000 mfcc feature vector and 2000 energy feature vector. We finally compute the distance of these feature vectors from the codebooks saved before. Depending upon the minimum distance measure, we finally annotate the input audio file with the corresponding genre name.

2.2.2.1 Results

We developed a database comprising of about 2000 audio files from 9 different genres. These audio files are downloaded from the website *www.magnatune.com* [22], which has been a source of audio database for audio processing over the past decade. We have used roughly 120 audio file per genre of 30 second each (excluding initial 30 seconds of the audio file) during the training phase. This accounts to almost 60 minutes of training dataset. While testing we have used about 80 audio files per genre. This accounts to about 40 minutes of testing dataset.

Here are the results obtained:

	Ambient	**Blues**	**Classical**	**Electronic**	**Folk**	**Jazz**	**New Age**	**Rap**	**Rock**
Ambient	**64**	0	12	6	6	6	4	2	0
Blues	0	**76**	0	6	2	6	0	0	10
Classical	0	0	**96**	0	0	0	2	2	0
Electronic	0	0	2	**94**	0	2	2	0	0
Folk	0	2	6	0	**88**	2	2	0	0
Jazz	0	0	0	6	0	**88**	0	6	0
New Age	2	3	15	0	3	10	**67**	0	0
Rap	0	1	0	5	0	0	0	**90**	4
Rock	0	0	0	2	2	5	0	6	**85**

Figure 9. Confusion Matrix using MFCC's as feature vector.

	Ambient	**Blues**	**Classical**	**Electronic**	**Folk**	**Jazz**	**New Age**	**Rap**	**Rock**
Ambient	**74**	0	10	4	4	4	4	0	0
Blues	0	**80**	0	4	0	6	0	0	10
Classical	0	0	**96**	0	0	0	0	4	10
Electronic	0	0	0	**96**	0	0	4	0	0
Folk	0	2	4	0	**92**	0	2	0	0
Jazz	0	0	0	4	0	**90**	0	6	0
New Age	3	0	13	0	2	7	**75**	0	0
Rap	0	2	0	5	0	0	0	**90**	3
Rock	0	0	0	5	0	1	0	6	**88**

Figure 10. Confusion Matrix using Log-Compressed Energies as feature vector.

CHAPTER 3
Automatic Music Organization Systems

The availability of large music collections calls for ways to efficiently access and explore them. We present an approach which combines descriptors derived from audio analysis with meta-information to create different views of a collection. Such views can have a focus on timbre, rhythm, artist, style or other aspects of music. For each view the pieces of music are organized on a map in such a way that similar pieces are located close to each other. The maps are visualized using an Islands of Music metaphor where islands represent groups of similar pieces.

We demonstrate our approach on a small collection using a meta-information-based view and two views generated from audio analysis, namely, beat periodicity as an aspect of rhythm and spectral information as an aspect of timbre.

3.1 Introduction

Music is magic. It influences our emotions. It has the power to make us happy or sad, just as it can make us relaxed or aggressive. Often it is associated with some very special moments in our lives. Moreover, music is an important part of our cultures and identities. However, the most fascinating aspect of music might be the fact that the annual turnover for record sales only within the US has a magnitude of several billion USD.

This huge industry would not exist without its customers, who are always looking for something new to listen to. There are many ways in which customers find their desired products. For example, one way is to listen around. Customers might listen to what is being played on the radio or to what friends are listening to. However, this type of search is restricted to the subjective taste of others. Furthermore, it might take a while until a new release reaches ones ears.

Another approach is to follow the development of artists, who have been appreciated in the past, assuming that their work will also be appreciated in the future. However, this kind of search does not include unknown artists or newcomers. Customers might also follow the development of a genre like Jazz, Hip Hop, Classic or Funk. Relying on the classification skills of other people it is possible to search music stores for new releases in certain genres. However, classifying music into a limited number of genres is not an easy task. Lots of music is located somewhere in between many genres.

The tool presented in this report is meant to help customers find music without limiting the search to a specific genre or artist. This tool is based on a metaphor of geographic maps. Genres of music are represented by islands and continents. Similar genres are located close together and might even be connected through some land passage. On these islands there might be some further sub-genres that are represented as mountains and hills. Again these sub-genres might be more or less similar to each other and are arranged accordingly.

The mountains and hills on the map are labelled with words that describe certain attributes of the associated genre, for example the type of rhythm is described rather than using words like Pop, Jazz or Classical. The pieces of music in the collection are placed on the map according to their genre or sub-genre. Most of the music will be located around the mountains. However the few located in the valleys between typical genres might be the most interesting ones. The user can listen to the music by clicking on its representation on the map and can explore island after island according to his or her musical

taste. Furthermore, music known to the user can be used as landmarks, to identify interesting regions on the map.

The maps with the islands of music could easily be placed on a web page of an internet store. Or they could be used in any conventional music store. Simple earphones and a touch screen monitor connected to a server would be sufficient. These maps could also be applied to digital libraries containing music, or simply at home to organize the private music collection. They could reveal some interesting properties of the inherent structure of the music collection that might not have been obvious before.
The technical requirements to develop music maps have only recently been met by the tremendous increase in computational power as well as the availability of affordable large storage. Now it is possible to handle the huge amount of data within music collections and to do the complex calculations leading to the music maps described above within reasonable time.

3.1.1 Scope and Overview

This report explores two main aspects related to music maps. One is how to compute the similarity of two pieces of music, so a whole music collection can be organized accordingly. The second aspect is how to present this information to the user in an intuitive way. The main goal of this work is to demonstrate the possibility of building a system, which enables efficient exploration of unknown music collections, given only the raw pieces of music without any Meta information.

This work uses a music collection of 77 pieces with a total length of about 6 hours to illustrate and evaluate the methods. A detailed list of all pieces of music, their authors and titles can be found in the Appendix. The music collection consists of a mixture of pieces of music from different genres. Most of these pieces are well known so that the reader can easily verify the presented results.

Related work can be found in *Chapter 2*. In particular the fields of content-based music analysis and approaches based on the Self-Organizing Map are discussed. Work related to details on psychoacoustics, clustering algorithms, and the visualization and automatic summarization of the results is presented in the corresponding chapters.

In *Chapter 3* the methods used to extract relevant features, which enable the computer to compare two pieces of music, are presented. These features are derived from the low-level raw audio signal without any additional Meta information. Based on psychoacoustic findings, features are constructed which reflect the dynamic and rhythmic properties of music. All feature extraction steps are illustrated using pieces of music from the music collection.

Chapter 4 deals with approach used to combine and analyze the extracted features. In *Chapter 5* a novel method to visualize clusters in a Self-Organizing Map is presented along with methods to give automatic summaries for groups of music. This approach is able to describe pieces of music with different lengths. The method is evaluated with the music collection

Finally in *Chapter 6* this work is concluded. A summary and further work together with interesting directions, with possibly very promising results are discussed. We end the report ends with References and Appendix.

3.2 Related Work

Music has been analyzed since the ancient Greeks. Pythagoras is credited with recognizing that strings whose lengths are related as the ratio of small integers sounds good when plucked at the same time. Since then a lot of research has been conducted and very sophisticated models and systems have been developed.

In the scope of this work especially systems which are designed to search for music based on its content are interesting, since this is the main motivation for this thesis. Section 2.1 reviews the literature on content-based music retrieval and section 2.2 focuses on approaches using the SOM algorithm. Work related to details on psychoacoustics, clustering algorithms, and the visualization and automatic summarization of the results is presented in the respective chapters.

3.2.1 Content-Based Music Retrieval

There are several possibilities to search for music based on its content. One is to use metadata information consisting of descriptions which have manually been assigned to each piece of music. These descriptions can be as simple as the name of the piece of music, but also more complex like an assignment to a specific genre or a verbal description of the music. The necessary standards are provided, for example, by the MPEG 7 standard [1]. A system based on MPEG 7 to compare sounds has, for example, been presented by [2]. Often pieces of music have lyrics, thus another possibility would be to apply methods from text document retrieval to search for music. Such systems are currently in use, for example, *BigLyrics.com* and *LyricCrawler.com* are two of the currently biggest music lyric search engines on the web. Using the lyrics as descriptions of the music it is possible to create an interface to allow an exploration of music archives where pieces of music with similar lyrics are located close to each other on a 2-dimensional map display using methods which have been developed mainly for text document collections, such as the SOMLib [3] or the WebSOM [4].

However, not always metadata is available and the metadata available might be erroneous, incomplete or inaccurate due to the deficiencies of manual labor. Likewise song lyrics are not always available; speech recognition systems only have a limited capability of extracting the lyrics automatically. And finally most music is available in MP3 or other similar formats rather than in MIDI. Thus content-based systems which directly analyze the raw music data (acoustical signals) have been developed. An overview of systems analyzing audio databases was presented by Foote [5]. However, Foote focuses particularly on systems for retrieval of speech or partly-speech audio data. Several studies and overviews related to content-based audio signal classification are available (e.g. [6]), however, they do not treat content-based music classification in detail.

Other approaches (e.g. [7, 8]) are based on methods developed in the digital speech processing community using *Mel Frequency Cepstral Coefficients* (MFCCs). MFCCs are motivated by perceptual and computational considerations, for example, instead of calculating the exact loudness sensation only decibel values are used. Furthermore the techniques appropriate to process speech data are not necessarily the best for processing music. For example, the MFCCs ignore some of the dynamic aspects of music. Recently Scheirer [9] presented a model of human perceptual behavior and briefly discussed how his model can be applied to classifying music into genre categories and performing music similarity-matching. However, he has not applied his model to large scale music collections. The collection he uses consisted of 75 songs from each of which he selected two 5-second sequences.

3.2.2 Approaches using Self-Organizing Maps (SOM)

This work is built upon the work of Fruhwirth and Rauber [10], who have shown that it is possible to cluster and organize music using neural networks. In their work they extract features from MP3 files which enable a self-organizing map (SOM) [11] to learn the inherent structure within a music collection. The feature extraction process consists of several steps.

They first transform the audio signals into the frequency domain using a fast Fourier transformation (FFT) with about 20 millisecond windows. In the frequency domain they select 17 frequencies for further processing. They split each piece of music into 5-second sequences. They remove the first and the last sequences to avoid fade-in and fade-out effects. From the remaining they select a subset using only every second to third sequence. Each frequency band from the selected sequences is then transformed into the frequency domain yielding 256 coefficients. They combine these 256 values for the 17 bands in a 4352-dimensional vector representing a 5-second sequence. These vectors reflect the dynamic properties of the selected frequencies.

A SOM is used to organize the large number of 5 second sequences on a 2-dimensional map in such a way that similar sequences are located close to each other. The different sequences of one piece of music might be scattered across the map if it contains a lot of variations. To get the pieces together again another feature vector is created using the information on where the different sequences of one piece of music are located. With this information another SOM is trained which organizes the pieces of music on a 2-dimensional map.

This work follows some of the proposals for further work presented by Fruhwirth and tries to combine the well working approach with psychoacoustics methods to improve the performance. An overview of psychoacoustics can be found in [12]. Psychoacoustics deals with the relationship of physical sounds and the human brain's interpretation of them.

3.3 Feature Extraction

Music with duration of 5 minutes is usually represented by 13 million values. These values describe the physical properties of the acoustical waves, which we hear. When analyzing this data it is necessary to remove the irrelevant parts and emphasize the important features. The extraction of these features from the raw data is the most critical part in the process of creating a content-based organization in a music collection. If it were possible to extract one single feature that directly indicates which genre a piece of music belongs to, everything else would be trivial? Good features should be intuitively meaningful, based on psychoacoustic findings, and robust towards variations which are insignificant to our hearing sensation. Furthermore, they should lead to an organization of the music collection that makes sense and not be too expensive to compute.

It is necessary to consider computational aspects because the raw data of even small music collections easily consumes several gigabytes of storage. A detailed analysis of all this information and all its possible meanings would be computationally prohibitive. It thus is necessary to reduce the amount of information to what is relevant in respect to the overall goal, which is to organize music according to its genre. These genres are not clearly defined and different people might assign the same piece of music to different genres. However, there are some attributes of the raw data, which definitely do not determine the genre. For example, removing the first second of a piece of music does not change its genre, but the raw data compared bit wise will be completely different. Generally the duration of a piece of music is not relevant. Neither does a particular melody define a genre. The same melody can be interpreted in different genre styles just as different melodies might be members of the same genre. Likewise, the number of instruments involved plays a minor role in defining the genre.

One of the attributes that is rather typical for a genre is its rhythm which is why this work primarily focuses on the dynamics of music, and in particular on the fluctuation strength [13] of the specific loudness per critical-band [14]. The following sections describe the feature extraction steps starting with the raw data, which is transformed from the time-domain to its frequency-domain representation. In the frequency-domain several transformations are applied to obtain the specific loudness per critical-band. Based on the specific loudness per critical-band the loudness fluctuation in a time interval of about 6 seconds is analyzed and an image in the dimensions of critical-band, modulation frequency, and fluctuation strength is created. To this image gradient filters (for edge detection) and Gaussian filters (for smoothening) are applied to emphasize important characteristics and remove insignificant ones. The modified fluctuation strength is used as final feature for the clustering algorithms.

3.3.1 Loudness Sensation

Loudness belongs to the category of intensity sensations. The loudness of a sound is measured by comparing it to a reference sound. The 1 kHz tone is a very popular reference tone in psychoacoustics, and the loudness of the 1 kHz tone at 40dB is defined to be 1 Sone. A sound perceived to be twice as loud is defined to be 2 sone and so on. To calculate the loudness sensation from raw audio data several transformations are necessary. The raw audio data is first decomposed into its frequencies using a discrete Fourier transformation. These frequencies are bundled according to the nonlinear critical-band rate scale (bark). Then spectral masking effects are applied before the decibel values are calculated. The decibel values are transformed to equal loudness levels (phon) and finally from these the specific loudness sensation is calculated (sone). At the end of this section each transformation is illustrated using examples from the music collection used for the experiments conducted for this work, as well as using some artificially generated sinusoidal signals

3.3.1.1 Discrete Fourier Transform (DFT)

Complex acoustical signals consist of several waves with different frequencies and amplitudes. The inner ear *(cochlea)* of humans decomposes the incoming acoustical waves into separate frequencies. The energy of different frequencies is transferred to and concentrated at different locations along the basilar membrane. Thus, it is appropriate to transform the PCM data into the frequency domain before analyzing it further. This can be achieved using, for example, *Fourier Transformations*. Alternatives include *Wavelets,* but are not considered in this work.

In this subsection only the most important characteristics are summarized. A more detailed description can be found, for example, in [15]. One of the aspects of music is that the frequencies change continuously; however, within very short time frames the frequencies are approximately constant. These very short sequences can be seen as fundamental building blocks of music. Thus, a piece of music can be described with subsequent frequency patterns, each representing a time quantum. A common choice for this interval is 20ms. The music data used for this work is sampled at 22050 Hz. To optimize the FFT the number of samples N should be a power of 2. However, this does not mean that it is necessary to have N samples. Shorter signals can be padded with zeros. Using 23ms time frames corresponds to 512 samples and results in a frequency resolution of about 43Hz in a range from 0 to 11 kHz. A 50% overlap between the windows is used. Notice that the sum of the two overlapping Hanning windows at any point always equals 1. A 50% overlap increases the time resolution by a factor 2 to about 12ms.

The calculations are implemented as follows. The raw audio data is given in vectors y(t) of length N corresponding to 23ms at the time frame t. The power spectrum matrix P (n; t), where n is the index for the frequency and t for the time frame can be calculated using,

$$\begin{aligned} y'_t &= W_N\, y_t, \\ Y_t &= fft(y_t), \text{ and} \\ P(n,t) &= |Y_t(n)|^2\, \frac{1}{N} \end{aligned} \qquad \text{..... (1)}$$

The index n ranges from 1 to N/2+1. The matrix W_N contains the Hanning function weights for N points on the diagonal with zeros elsewhere where N = 512 at 11 kHz. The *fft* function is taken from the FFTW library. The data after this feature extraction step basically still has the same size. While the discrete Fourier transformation yields 256 values for 512 sample values, the 50% overlap increases the amount of data by 2.

3.3.1.2 Critical Bands

So far a piece of music is represented by a frequency snapshot every 12ms. These have one value every 43Hz starting at 0Hz up to 11 kHz, where each value represents the power of the respective frequency. As stated previously, the inner ear separates the frequencies, transfers, and concentrates them at certain locations along the basilar membrane. The inner ear can be regarded as a complex system of a series of band-pass filters with an asymmetrical shape of frequency response. The center frequencies of these band-pass filters are closely related to the critical-band rates. Where these bands should be centred or how wide they should be, has been analyzed throughout several psychoacoustic experiments [12]. While we can distinguish low frequencies of up to about 500Hz well, our ability decreases above 500Hz with approximately a factor of 0.2f, where f is the frequency. This is shown in experiments using a loud tone to mask a quieter one. At high frequencies these two tones need to be rather far apart regarding their frequencies, while at lower frequencies the quiet tone will still be noticeable at smaller distances. In addition to these masking effects the critical-bandwidth is also very closely related to just noticeable frequency variations. Within a critical-band it is difficult to notice any

variations. This can be tested by presenting two tones to a listener and asking which of the two has a higher or lower frequency.

Since the critical-band scale has been used very frequently, it has been assigned a unit, the *bark*. The name has been chosen in memory of Barkhausen, a scientist who introduced the phon to describe loudness levels for which critical-bands play an important role. Figure 2 shows the main characteristics of this scale. At low frequencies below 500Hz the critical-bands are about 100Hz wide. The width of the critical bands increases rapidly with the frequency. The 24th critical-band has a width of 3500Hz. The 9th critical-band has the center frequency of 1 kHz. The critical-band rate is important for understanding many characteristics of the human ear.

A critical-band value is calculated by summing up the values of the power spectrum within the respective lower $f_a(i)$ and upper $f_b(i)$ frequency limits of the i^{th} critical band. This can be formulated as:

$$B(i,t) = \sum_{n \in I(i)} P(n,t),\ I(i) = \left\{ n \,\middle|\, f_a(i) < f_{\text{res}}(n-1, 256, 1/11025) \le f_b(i) \right\} \quad \text{..... (2)}$$

Where *i*, *t*, *n* are indexes and B is a matrix containing the power within the *i*-th critical band at a specific time interval *t*. P is the matrix representing the power per frequency and time interval *t* obtained from Equation (1). Notice that P (1*; t*), which represents the power at 0Hz, is not used. While the critical-band rate is defined having 24 bands, only the first 20 are used in this work, since the highest frequencies in the data are limited to 11 kHz. The 256 power spectrum values are now represented by 20 critical-bands values. This corresponds to a data reduction by a factor of about 6.5.

3.3.1.3 Masking

As mentioned before, the critical-bands are closely related to masking effects. Masking is the occlusion of one sound by another sound. A loud sound might mask a simultaneous sound (simultaneous masking), or a sound closely following (post-masking) or preceding (pre-masking) it. Pre-masking is usually neglected since it can only be measured during about 20ms. Post-masking, on the other hand can last longer than 100ms and ends after about a 200ms delay. Simultaneous masking occurs when the test sound and the masker are present simultaneously. For this report the spreading function is used to estimate the effects of simultaneous masking across the critical-bands. The spreading function defines the influence of the *j*-th critical-band on the *i*-th and is calculated as:

$$S(i,j) = 15.81 + 7.5(i - j + 0.474) - 17.5\sqrt{1 + (i - j + 0.474)^2} \quad \text{..... (3)}$$

The spread critical-band rate spectrum matrix B_s is obtained by multiplying B with S as follows:

$$B_s(i,t) = \sum_{j=1}^{20} S(i,j) B(j,t), \text{ which is equivalent to} \quad \text{..... (4)}$$

$$B_s = SB.$$

The simultaneous masking asymmetrically spreads the power spectrum over the critical bands. The masking influence of a critical-band is higher on bands above it than on those below it.

3.3.1.4 Decibel

Before calculating sone values it is necessary to transform the data into decibel. The intensity unit of physical audio signals is sound pressure and is measured in *Pascal* (Pa). The values of the PCM data correspond to the sound pressure. It is very common to transform the sound pressure into *decibel* (dB). Decibel is the logarithm, to the base of 10, of the ratio between two amounts of power. The decibel

value of a sound is calculated as the ratio between its pressure and the pressure of the hearing threshold given by 20uPa. The sound pressure level in dB is calculated as

$$S_{dB} = 10\log_{10}\frac{p}{p_0} \qquad \text{..... (5)}$$

Where p is the power of the sound pressure, and p_0 is the power of the sound pressure of the hearing threshold. The power is calculated as the squared sound pressure. Parseval's theorem states that the power of the signal is the same whether calculated in the time domain or the frequency domain, so the dB values can be calculated for the spread critical-band matrix BS. A parameter to adjust is the reference value $p0$. Note that the influence of the hearing threshold p_0 on the decibel calculations is non-linear.

If the hearing threshold is too low insignificant sounds will become significant, on the other hand if it is too high significant sounds will become insignificant. Knowing that the sound pressure of the signals is digitized using 16 bit, it could be assumed that the most quiet, just noticeable power of the sound pressure level corresponds to 1 (or -1). When using $p_0 = 1$ the notation db (SPL) is used, where SPL stands for sound pressure level. The maximal decibel value for the energy at a certain frequency using db (SPL) is 96dB. However, the use of this assumption has led to sone values beyond the limit of damage risk in the experiments conducted for this report. This occurs because the energy of the frequencies are added together in the critical-bands and the masking function used is additive. The problem with decibel values beyond the limit of damage risk is that only equal loudness contours for levels below the limit are available, thus it is not possible to calculate accurate phon values (see next section), which have a non-linear correlation to the decibel values. To be able to calculate the phon values the PCM amplitudes of the music collection were scaled so that all sounds are below the limit of damage risk. This corresponds to turning down the volume to a level at which the loudness of all music listened to, is within healthy ranges. The hearing threshold parameter p_0 was set to 1/0.35. The loudness matrix in decibel, L_{dB}, is calculated as follows,

$$\begin{aligned} \mathrm{B}'_s(i,t) &= \min(B_s(i,t), p_0), \text{ and} \\ \mathrm{L}_{\mathrm{dB}}(i,t) &= 10\log_{10}\frac{1}{p_0}\mathrm{B}'_s(i,t) \end{aligned} \qquad \text{..... (6)}$$

Note that the first step is made in order to avoid the logarithm of zero.

3.3.1.5 Phon

The relationship between the sound pressure level in decibel and our hearing sensation measured in sone is not linear. The perceived loudness depends on the frequency of the tone. The phon is defined using the 1 kHz tone and the decibel scale. For example, a pure tone at any frequency with 40 phon is as loud as a pure tone with 40dB at 1 kHz. We are most sensitive to frequencies around 2 kHz to 5 kHz. The hearing threshold rapidly raises around the lower and upper frequency limits, which are respectively about 20 Hz and 16 kHz.

Although the equal loudness contours are obtained from experiments with pure tones, they are frequently applied to calculate the specific loudness of the critical band rate spectrum. The loudness matrix in phon, L_{phon}, can be calculated using the equal loudness contour matrix C_{elc} and the corresponding phone values to each contour c_{phon} = [3; 20; 40; 60; 80; 100]. $C_{elc}(i; j)$ contains the decibel values of the j-th loudness contour at the i-th critical-band. Values in between two equal loudness contours are interpolated linearly, as follows:

$$
\begin{aligned}
&L'_{dB}(i,t) = \max(L_{dB}(i,t), C_{elc}(i,1)),\\
&level_{i,t} = \arg\min_{j}(L'_{dB}(i,t) < C_{elc}(i,j)),\\
&r_{i,t} = \frac{L'_{dB}(i,t) - C_{elc}(i, level_{i,t} - 1)}{C_{elc}(i, level_{i,t}) - C_{elc}(i, level_{i,t} - 1)}, \text{ and}\\
&\mathrm{L}_{\mathrm{phon}}(i,t) = c_{phon}(level_{i,t} - 1) + r_{i,t} c_{phon}(level_{i,t})
\end{aligned}
\qquad \text{..... (7)}
$$

Note that $L_{phon}(i; t)$ can only be calculated if there exists an equal loudness contour j in $C_{elc}(i; j)$ so that $C_{phon}(j) > L_{phon}(i; t)$. The highest decibel level used here is 100 dB and p_0 (see above) is adjusted manually so all sounds are below this limit.

3.3.1.6 Sone

Finally, from the loudness level L_{phon} the specific loudness sensation L_{sone} per critical band, following [16], is calculated as,

$$
L_{sone}(i,t) = \begin{cases} 2^{\frac{1}{10}(L_{phon}(i,t) - 40)} & \text{if } L_{phon}(i,t) > 40 \\ \left(\frac{1}{40} L_{phon}(i,t)\right)^{2.642} & \text{otherwise.} \end{cases}
\qquad \text{..... (8)}
$$

For low values up to 40 phon the sensation raises slowly until it reaches 1 sone at 40 phon. Beyond 40 phon the sensation increases at a faster rate. The highest values that occurred in the experiments conducted for this thesis are below 60 sone, due to the adjustment of the threshold in quiet $p0$.

3.3.1.7 System Architecture for Sonogram Calculation

Prior to Sonogram Extraction, each mp3 (stereo) file is converted to wav (mono) file. Also each audio file is segmented into 6-second sequences. For feature extraction, the first and the last 6-sec sequence of the audio song are left out to eliminate the fade-in and fade-out effect, also only every third 6-sec sequence is chosen for feature extraction. Now every 6-sec sequence is given as input to the block diagram below for Sonogram extraction:

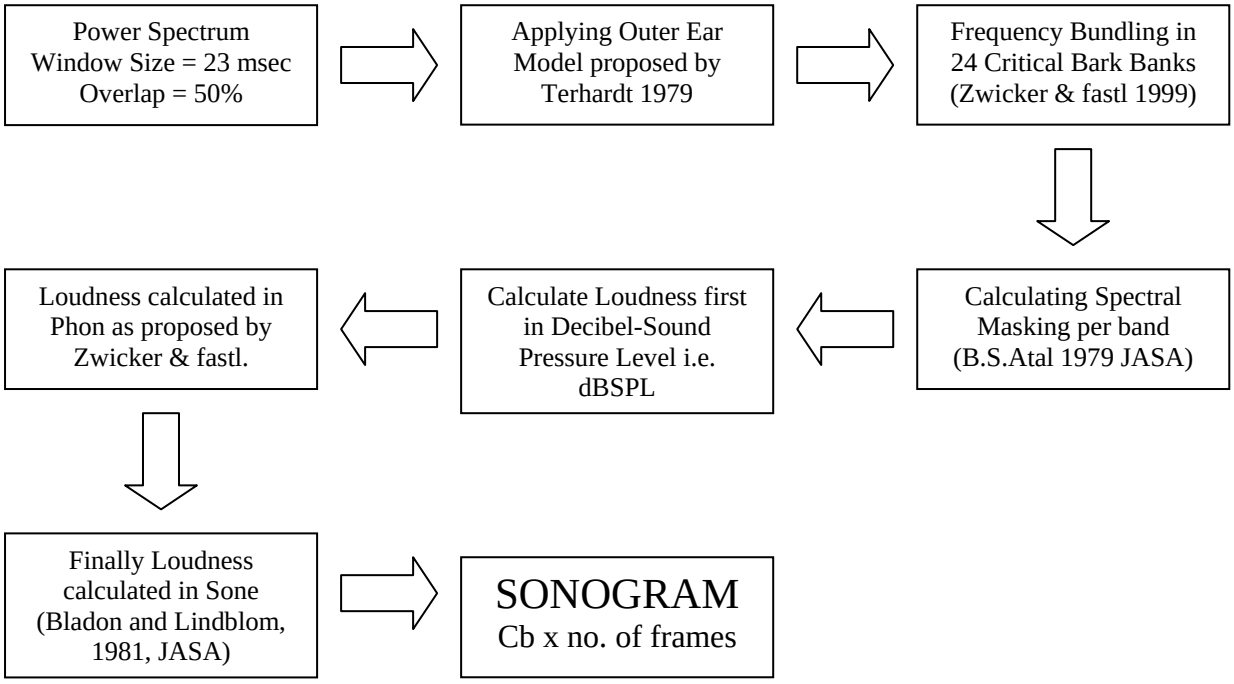

Figure 11. Flow Diagram for Sonogram Calculation

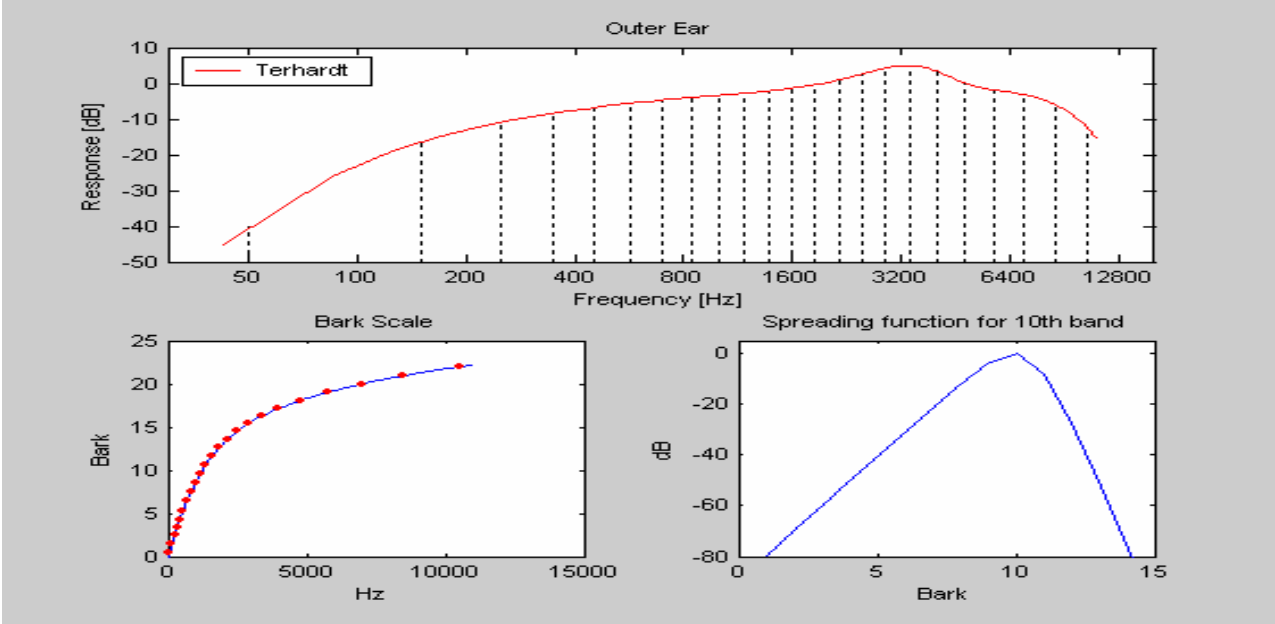

Figure 12. The main characteristics of the sone model. The upper subplot shows the width of the critical-bands and the outer-ear model. The lower left shows the relationship between the Bark-scale and Hz. The lower right shows the spectral masking function. Some things to notice: The Bark-scale is linear up to about 500Hz. The spectral masking is not symmetric. Higher tones are masked stronger by lower ones than vice versa. The outer-ear is most responsive to frequencies around 3-4 kHz.

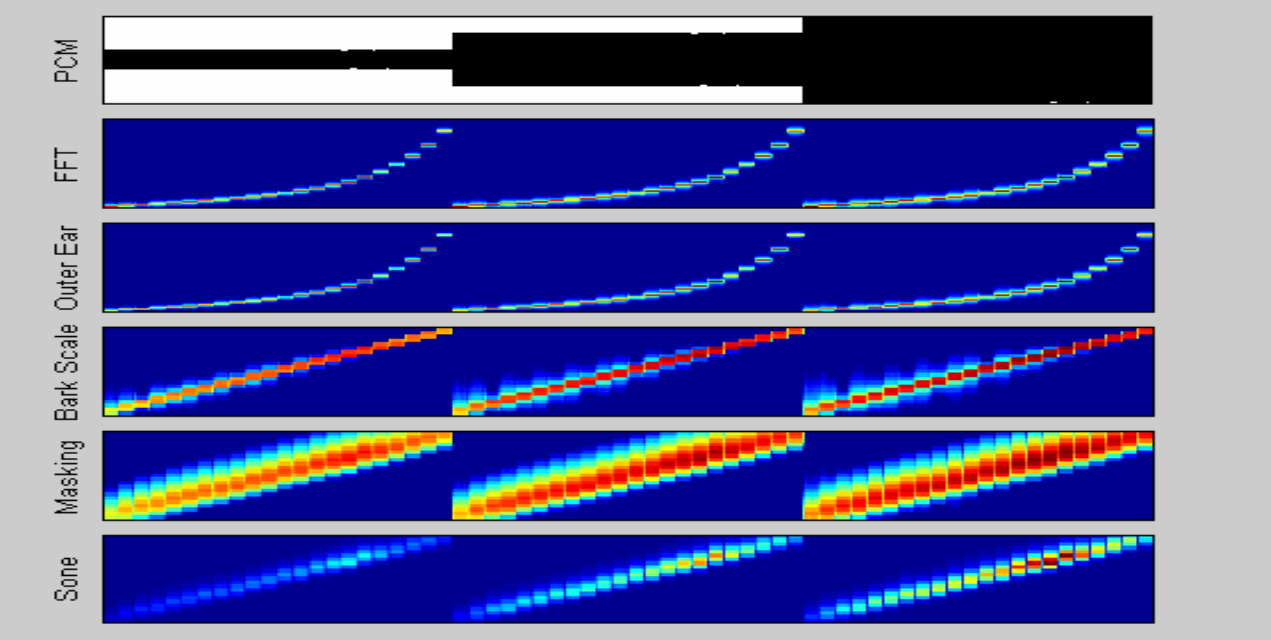

Figure 13. The characteristics of the sone model can be demonstrated with a simple sine-tone. The tone is generated at different frequencies (center frequencies of the critical-bands) with fixed amplitude (3 times, each time the fixed amplitude is increased). Note the difference between the linear frequency scale (used in the outer-ear and FFT representations) and the Bark-scale (the lower 3 subplots). The sinusoids in each sweep from low to high have the same amplitudes, they are not perceived equally loud. This pitch dependent loudness sensation is reflected in the model (in contrast to the MFCC model).

3.3.1.8 System Architecture for Spectrum Histogram

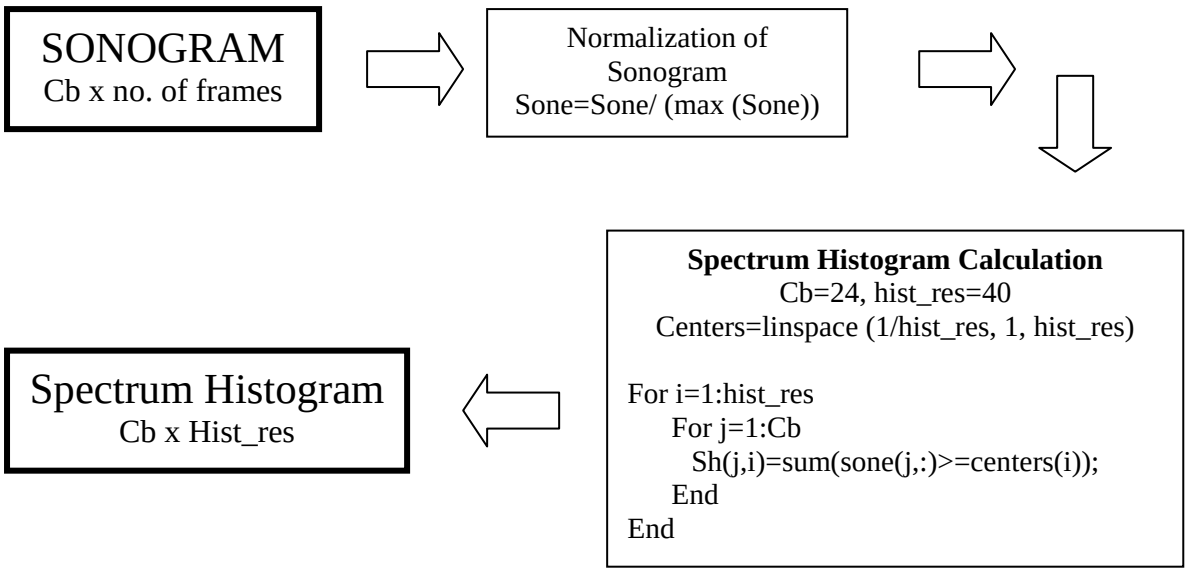

Figure 14. Flow Diagram for Spectrum Histogram calculation

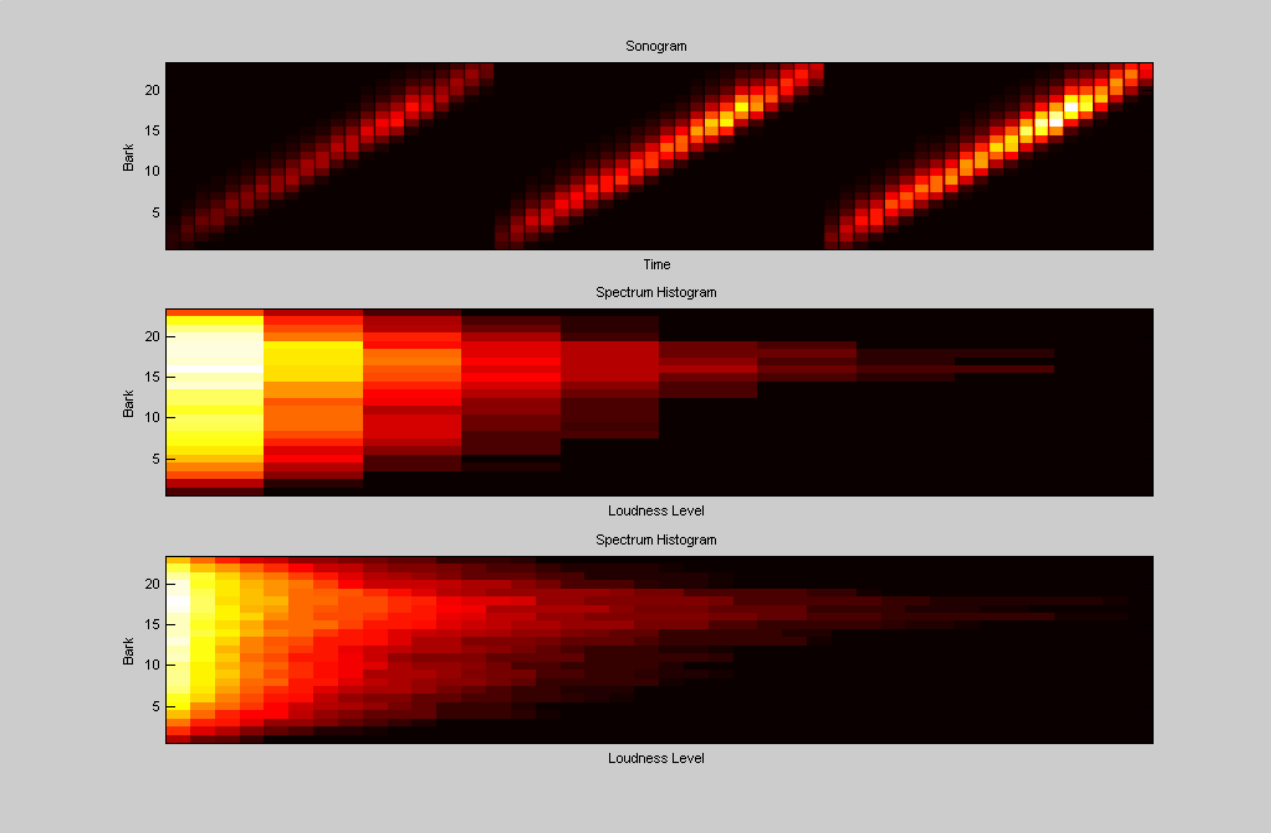

Figure 15. Same test sound as in the examples above. The spectrum histogram counts how often a loudness level was exceeded in each critical-band. Two spectrum Histogram are shown with hist_res=10 and hist_res=25 respectively.

3.3.2 Dynamics

So far each song is represented by several 6-second sequences of the specific loudness sensation per critical-band, $L_{sone}(i; t)$. It would be possible to use $L_{sone}(i; t)$ to calculate similarities between the data. One option would be to compare two sequences La_{sone} and Lb_{sone} point-wise, i.e. comparing $La_{sone}(i; t)$ and $Lb_{sone}(i; t)$ for all i and t. The result might be quite surprising. For example, shifting *Rock DJ* by only 40ms would result in a huge difference to the un-shifted sequences - although they sound the same. The same problem would occur for any of the other sequences presented in the previous section. Thus, the final representation of the data must be invariant to time shifts.

As mentioned before, the aim is to gather information on the dynamics of a sequence. Weil [17] and Fruhwirth have used the Fourier transforms of the activities in the frequency bands, which are also used here. Ellis [18] has shown that using a similar concept it is possible to predict the pitch of a sound. Ellis analyzed periodically recurring peeks in the loudness of a frequency band by calculating the autocorrelation. The main difference between Weil, Fruhwirth and Ellis is the frequency ranges they analyze. While Ellis analyzes patterns with periods of about one millisecond (which corresponds to frequencies up to 1 kHz), Fruhwirth limits his investigation to frequencies up to 25Hz. Weil uses a similar spectrum as Fruhwirth, though limits his analysis to 15Hz.

3.3.2.1 Amplitude Modulated Loudness

The loudness of a critical-band usually rises and falls several times. Often there is a periodical pattern, also known as the rhythm. At every beat the loudness sensation rises, and the beats are usually very accurately timed. The loudness values of a critical-band over a certain time period can be regarded as a signal that has been sampled at discrete points in time. The periodical patterns of this signal can then be assumed to originate from a mixture of sinusoids. These sinusoids modulate the amplitude of the

loudness, and can be calculated by a Fourier transform. An example might illustrate this. To add a strong and deep bass with 120 *beats per minute* (bpm) to a piece of music, a good start would be to set the first critical-band (bark 1) to a constant noise sensation of 10 sone. Then one could modulate the loudness using a sine wave with a period of 2Hz and amplitude of 10 sone.

The modulation frequencies, which can be analyzed using the 6-second sequences and time quanta of 12ms, are in the range from 0 to 43Hz with an accuracy of 0.17Hz. Notice that a modulation frequency of 43Hz corresponds to almost 2600bpm. The modulation amplitude delta L_i (*n*) with the frequency of the *i*-th critical-band is calculated as follows:

$$\Delta L_i = fft(L_{some}(i, 1....511)), \qquad \text{..... (8)}$$

Where $L_{some}(i;\ 1 \ldots. 511)$ is a 6-second sequence of the *i*-th critical-band of any piece of music. The *fft* function is the same as in Equation (1). Since there are only 511 values for the FFT, the signal is padded with one zero.

3.3.2.2 Fluctuation Strength

The amplitude modulation of the loudness has different effects on our sensation depending on the frequency. The sensation of *fluctuation strength* [13] is most intense around 4Hz and gradually decreases up to a modulation frequency of 15Hz (cf. Figure 3.9). At 15Hz the sensation of *roughness* starts to increase, reaches its maximum at about 70Hz, and starts to decreases at about 150 Hz. Above 150 Hz the sensation of hearing *three separately audible tones* increases [12]. The fluctuation strength of a tone with the loudness delta *L*, which is 100% amplitude modulated with the frequency f_{mod} can be expressed by,

$$f_{flux}(\Delta L, f_{\mathrm{mod}}) \propto \frac{\Delta L}{(f_{\mathrm{mod}} / 4H_z) + (4H_z / f_{\mathrm{mod}})} \qquad \text{..... (9)}$$

The modulation amplitudes F(*i; n*) of the *i*-th critical-band are weighted according to the fluctuation strength sensation as follows,

$$F(i,n) = f_{flux}(|\Delta L_i(n+1)|, f_{res}(n)), \qquad \text{..... (10)}$$

3.3.2.3 System Architecture for Fluctuation Pattern (FP)

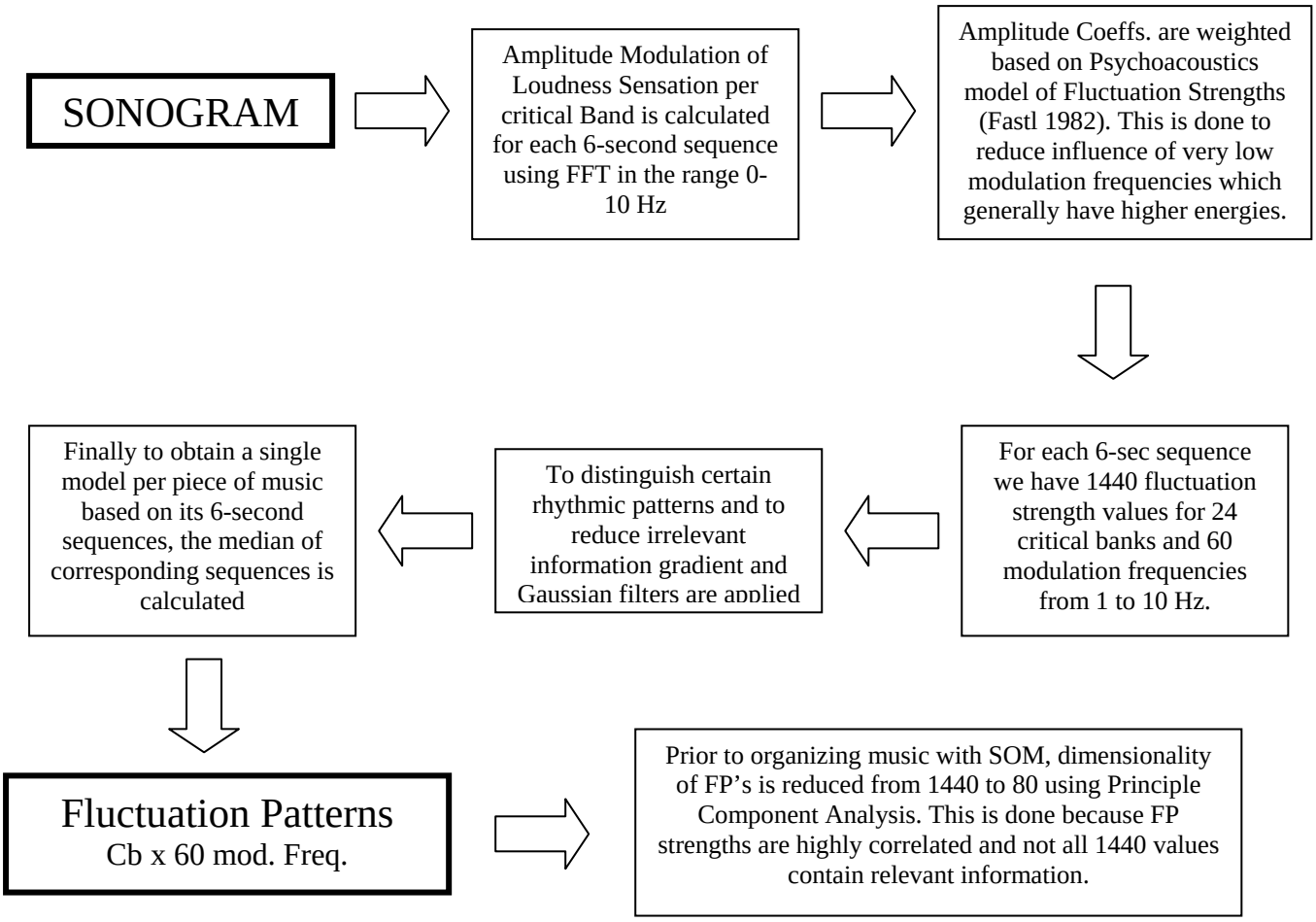

Figure 16. Flow Diagram for Fluctuation Pattern Calculation

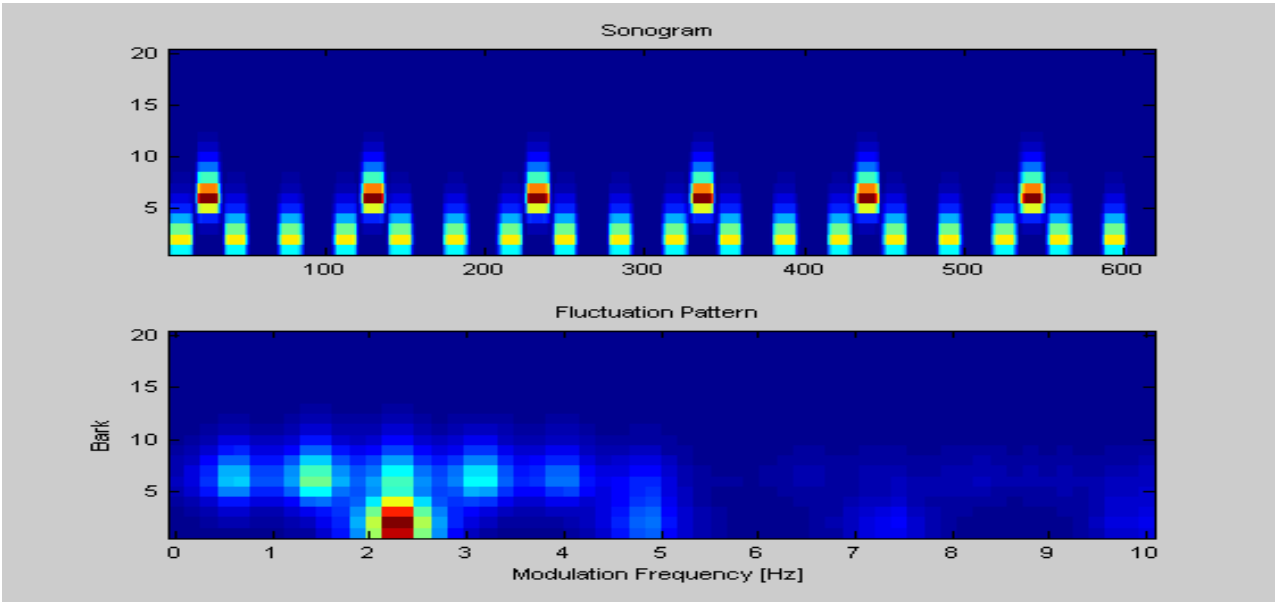

Figure 17. Final Result after computing the fluctuation pattern for a 6sec sequence.

3.3.3 Periodic Histograms

To obtain periodicity histograms we use an approach presented by Scheirer (1998) in the context of beat tracking. A similar approach was developed by Tzanetakis and Cook (2002) to classify genres. There are two main differences to this previous work. First, we extend the typical histograms to incorporate information on the variations over time which is valuable information when considering similarity. Second, we use a resonance model proposed by Moelants (2002) for preferred tempo to weight the periodicities and in particular to emphasize differences in tempos around 120 beats per minute (bpm). We start with the pre-processed data and further process it using a half wave rectified difference filter on each critical-band to emphasize percussive sounds. We then process 12 second windows (1024 samples) with 6 second overlap (512 samples). Each window is weighted using a Hann window before a comb filter bank is applied to each critical-band with a 5bpm resolution in the range from 40 to 240bpm. Then we apply the resonance model of Moelants (2002) with Beta = 4 to the amplitudes obtained from the comb filter. To emphasize peaks we use a full wave rectified difference filter before summing up the amplitudes for each periodicity over all bands.

That gives us, for every 6 seconds of music, 40 values representing the strength of recurring beats with tempos ranging from 40 to 240bpm. To summarize this information for a whole piece of music we use a 2-dimensional histogram with 40 equally spaced columns representing different tempos and 50 rows representing strength levels. The histogram counts for each periodicity how many times a level equal to or greater than a specific value was reached. This partially preserves information on the distribution of the strength levels over time. The sum of the histogram is normalized to one, and the distance between two histograms is computed by interpreting them as 2000-dimensional vectors in a Euclidean space.

Examples for periodicity histograms are given in Figure 4. The histogram has clear edges if a particular strength level is reached constantly and the edges will be very blurry if there are strong variations in the strength level. It is important to notice that the beats of music with strong variations in tempo cannot be described using this approach. Furthermore, not all 2000 dimensions contain information. Many are highly correlated, thus it makes sense to compress the representation using principal component analysis. For the experiments presented in this paper we used the first 60 principal components. A first quantitative evaluation of the periodicity histograms indicated that they are not well suited to measure the similarity of genres or artists in contrast to measures which use spectrum information. One reason might be that the pieces of an artist might be better distinguishable in terms of rhythm than timbre. However, it is also important to realize that using periodicity histograms in this simple way (i.e., interpreting them as images and comparing them pixel-wise) to describe rhythm has severe limitations. For example, the distance between two pieces with strong peaks at 60bpm and 200bpm is the same as between pieces with peaks at 100bpm and 120bpm.

3.3.3.1 System Architecture for Periodic Histogram

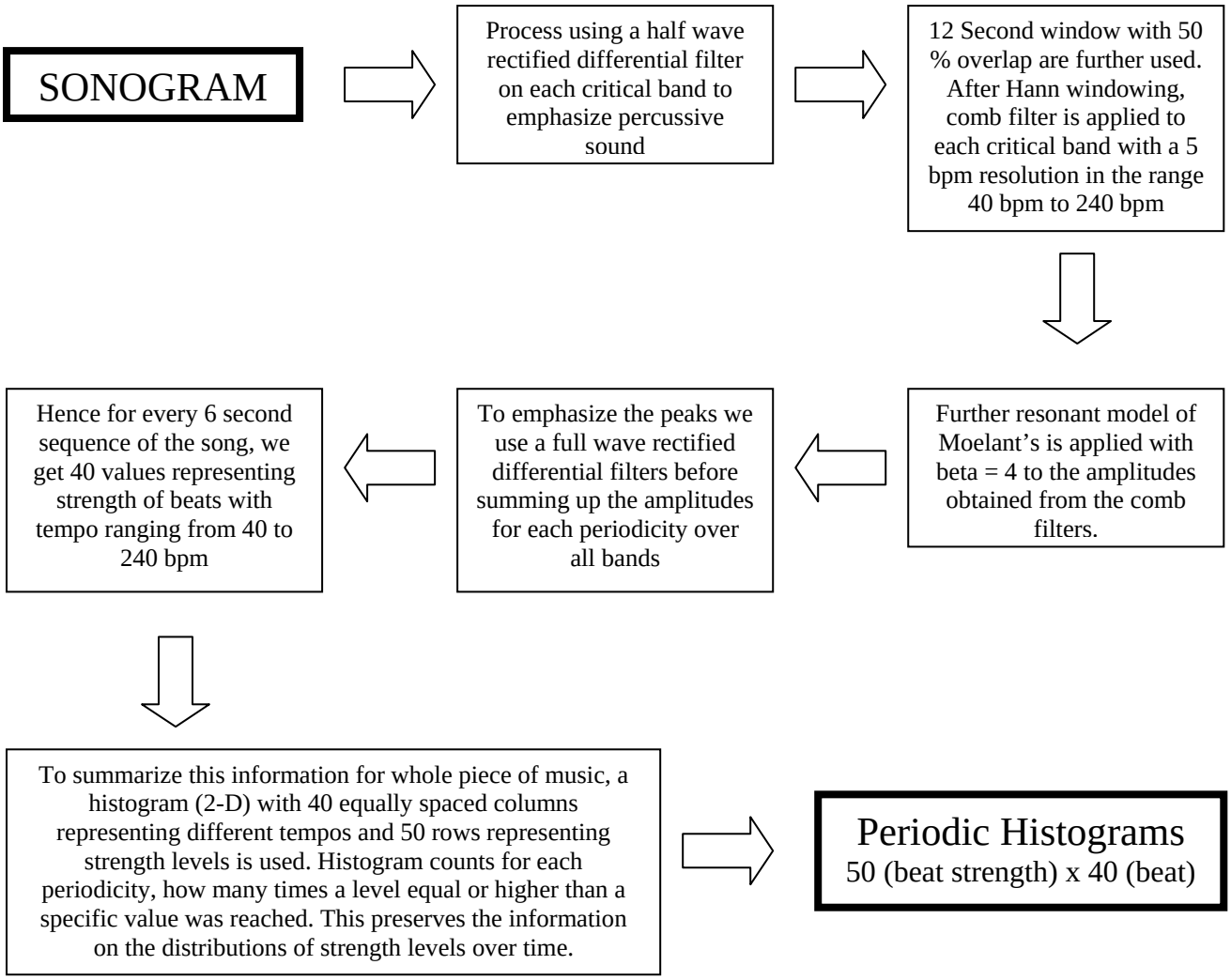

Figure 18. Flow Diagram for Periodic Histogram Calculation

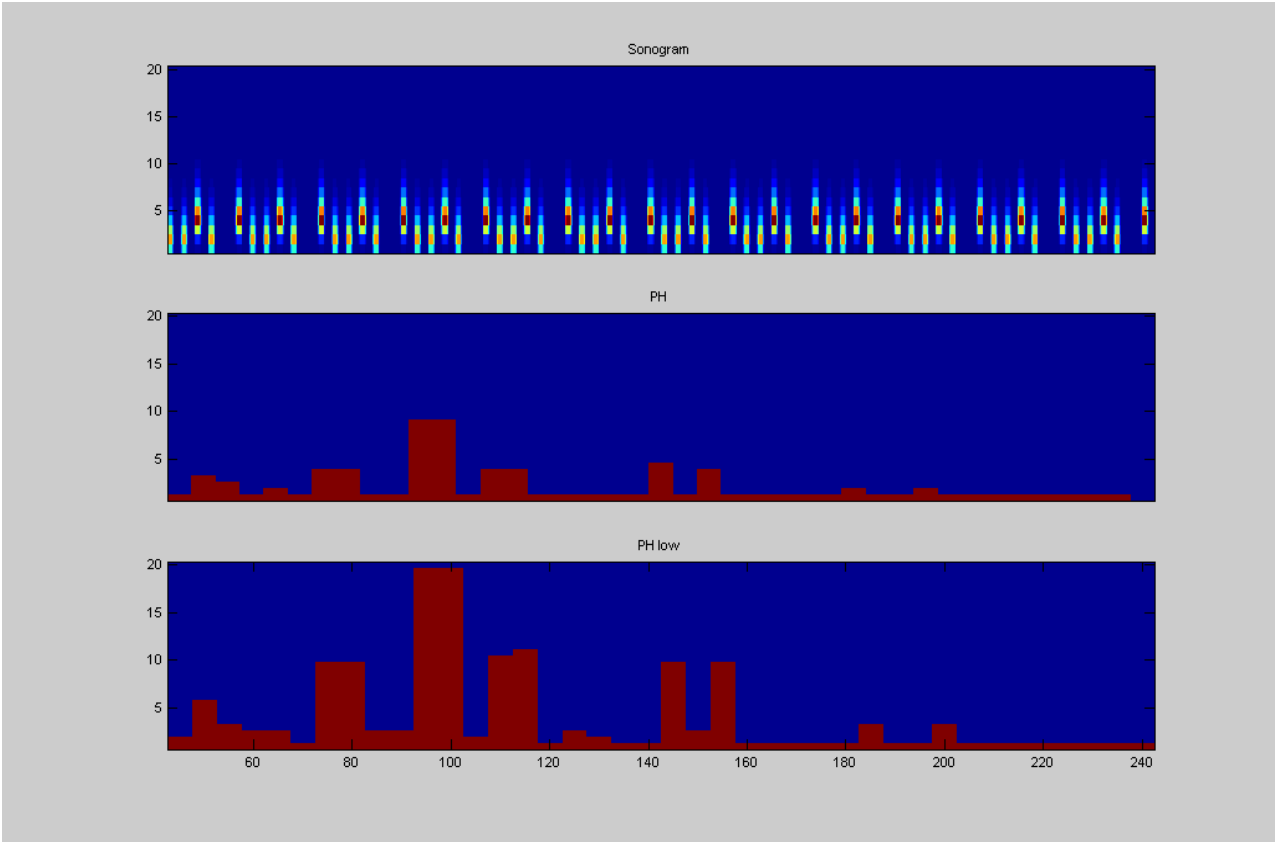

Figure 19. If the input is long enough, then the results from each 12sec frame are summarized in a histogram, counting for each periodicity (bpm) how often specific energy levels were exceeded.

3.4 Clustering

The previous chapter dealt with the extraction of features from the raw music data. In this chapter the music collection (Appendix B) is organized based on these features. The main tool used for this is the self-organizing map (SOM) algorithm, which is a clustering method with intuitive visualization capabilities. In the process of developing this thesis the SOM has been applied several times to support the evaluation of the feature extraction process and the SOM is the basis of the final user interface to the music collection presented in this thesis.

In Section 4.1 the SOM is described briefly. Applying the SOM in Section 4.2 a brief evaluation of the features extracted in Chapter 3 is presented and discussed. Section 4.3 deals with different approaches to represent one piece of music based on the representation of its sequences. And finally in Section 4.4 the chapter is summarized.

3.4.1 Self-Organizing Maps (SOM)

The goal of clustering data is to find groups (clusters) of data items that are similar to each other and different from the rest of the dataset. Clustering is a method to summarize the main characteristics of a dataset. For example, a music collection could be summarized by describing the groups (genres) it consists of. Each group could be described, for example, by the number and variation of its members. It might also be interesting to know the relationship between these groups. For example, a genre might have several sub-genres. For the purpose of clustering data several algorithms have been developed. A recent review can be found in [19]. One very frequently employed clustering algorithm is the SOM.

3.4.1.1 Background

The SOM was developed 1981 [20] as an artificial neural network, which models biological brain functions. Since then it has undergone thorough analysis [Koh01]. The algorithm and its variations have been employed several times in domains such as machine vision, image analysis, optical character recognition, speech analysis, and engineering applications in general. The SOM is a powerful tool that can be used in most data-mining processes especially in data exploration. Moreover, the SOM is very efficient compared to other non-linear alternatives such as the Generative Topographic Mapping [21], Sammon's Mapping [22], or generally Multi Dimensional Scaling. An ex- ample for the efficiency of the SOM is the WebSOM1 project.

3.4.1.2 The Batch SOM Algorithm

One of the variations of the original SOM is the batch SOM algorithm, which is significantly faster and has one parameter less to adjust [23]. Although the algorithm is different the architecture of the map is the same. The map consists of map units, which are ordered on a grid. Usually this grid is rectangular and 2-dimensional and is used to visualize the data. An example can be seen in Figure 5. Each of the map units is assigned to a reference vector, also known as model or prototype vector. This vector lies in the data space and represents the data items that are closest to it. Units, which are close to each other on the grid, also have similar model vectors and thus represent similar data.

The batch SOM algorithm consists of two steps that are iteratively repeated until no more significant changes occur. First the distance between all data items x_i and the model vectors m_j is computed and each data item i is assigned to the unit that represents it best C_i. V_j denotes the Vernoi set of data items which are best represented by unit m_j.

In the second step each model vector is adapted to better fit the data it represents. To ensure that each unit *j* represents similar data items as its neighbours, the model vector m*j* is adapted not only according to V*j* but also in regard to the Vernoi sets of the units in the neighborhood. Which units are considered to be neighbors and how much influence they have on the unit *j* is defined by a neighborhood function. A common choice for the neighborhood function *ht(j; k)* is a Gaussian function which is centered on m*j* and has a standard deviation $sigma_t$ which decreases with each iteration *t*. Assuming a Euclidean vector space, the two steps of the batch SOM algorithm can be formulated as

$$c_i = \arg\min_j \left\| x_i - m_j \right\|^2, and$$

$$m_j^* = \frac{\sum_i h_t(j, c_i) x_i}{\sum_{i'} h_t(j, c_{i'})}, \qquad \text{..... (11)}$$

Where m*$_j$ is the updated model vector. Although it is usually very unlikely it might occur that a data vector x*i* is equally closest to two or more model vectors. In this case randomly one of these model vectors should be chosen to be *Ci*. An efficient way to implement this is to first calculate the sum S*j* of all data vectors in a Vernoi set V*j* and then use them to calculate the weighted average,

$$V_j = \{ x_i \mid c_i = j \},$$

$$s_j = \sum_{x_i \in V_k} x_i, and$$

$$m_j^* = \frac{\sum_k h_t(j,k) s_j}{\sum_{k'} h_t(j,k') |V_{k'}|}$$

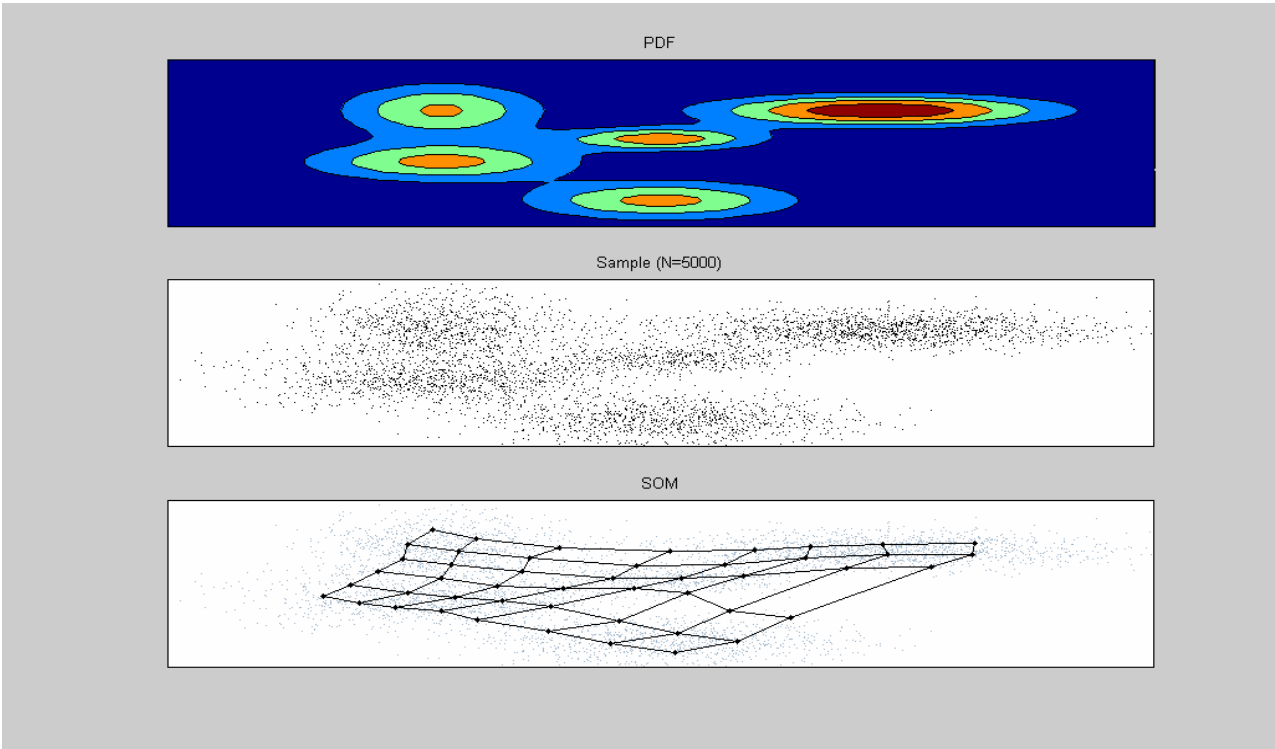

Figure 20. A test result after running SOM Algorithm on the above shown 5 clusters.

3.5 Visualization and Interface

In the previous Chapter SOMs labelled with the song identifiers were used to evaluate the music collection. Basically these maps could be used as user interfaces. Similar songs are located close together on the map and are identified by a string short enough to fit the width of a map unit. While these maps surely are a bigger help then a simple alphabetical list there are two major deficiencies.

The first is the lacking support trying to understand the cluster structure. Looking, for example, at Figure 10 it is rather difficult to recognize clusters on the first sight. Only after carefully studying the whole map it appears, for example, that there is a cluster of classical music in the lower left corner. Thus some visual support to identify clusters would be desirable. The second deficiency derives from the assumption that the music collection and the contained pieces of music are unknown to the user, thus any information on artists or titles are not very useful for the user. To a user a map labeled with unknown music titles, interprets or authors might not be much more useful than randomly generated text. By far more interesting for the user would be some text, which explains what type of music, is mapped to a map unit.

In the following sections methods are described which aim at creating a more intuitive user interfaces. In Section 5.1 the problem of visualizing cluster structure is addressed followed by section where important map areas are summarized and labeled. Both sections use the map presented in Figure 11 to illustrate the different possibilities.

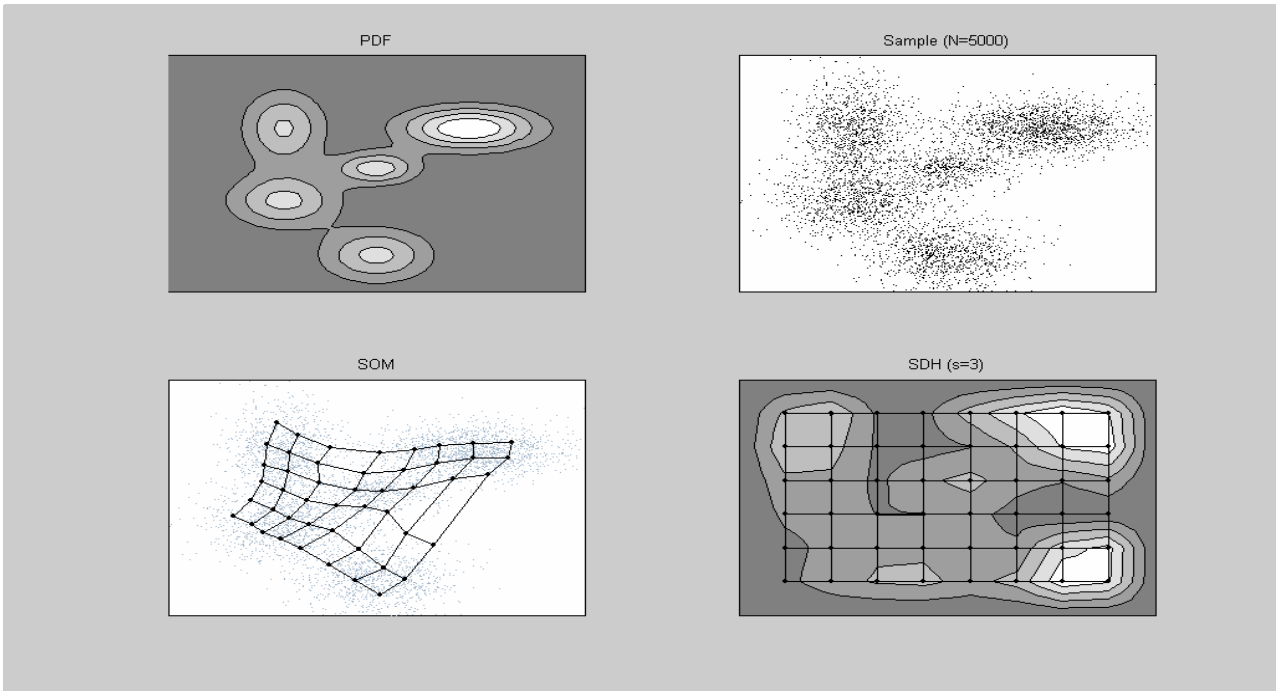

Figure 21. A Smooth Data Histogram representation of results obtained by SOM algorithm in Figure 10.

3.5.1 Islands

A useful visualization system should offer the user a good summary of the relevant information. To be effective the distinguishing features (e.g. position, form, and color) in the visual dimensions should be detectable effortlessly and quickly by the human visual system in the preattentive processing phase. In general the efficiency of visualization will depend on the domain, culture, and personal preferences of the users.

The results of the previous chapters can be visualized in several different manners, only a few will be discussed in this section. This report has been titled *Islands of Music* because the metaphor of islands is used to visualize music collections. The metaphor is based on islands, which represent groups of similar data items (pieces of music). These islands are surrounded by the sea, which corresponds to areas on the map where mainly outliers or data items, which do not belong to specific clusters, can be found. The islands can have arbitrary shapes, and there might be land passages between islands, which indicate that there are connections between the clusters. Within an island there might be sub-clusters. Mountains and hills represent these. A mountain peek corresponds to the centres of a cluster and an island might have several mountains or hills.

More accurately a simple approximation of the probability density function is visualized using a color scale which ranges from dark blue (deep sea) to light blue (shallow water) to yellow (beach) to dark green (forest) to light green (hills) to gray (rocks) and finally white (snow). The exact color scale represented by HSV (see e.g. [24]) values. The sea level has been set to 1/3 of the total color range, thus map units with a probability of less then 1/3 are under water. The visible effects might aid understanding the islands and the corresponding clusters better.

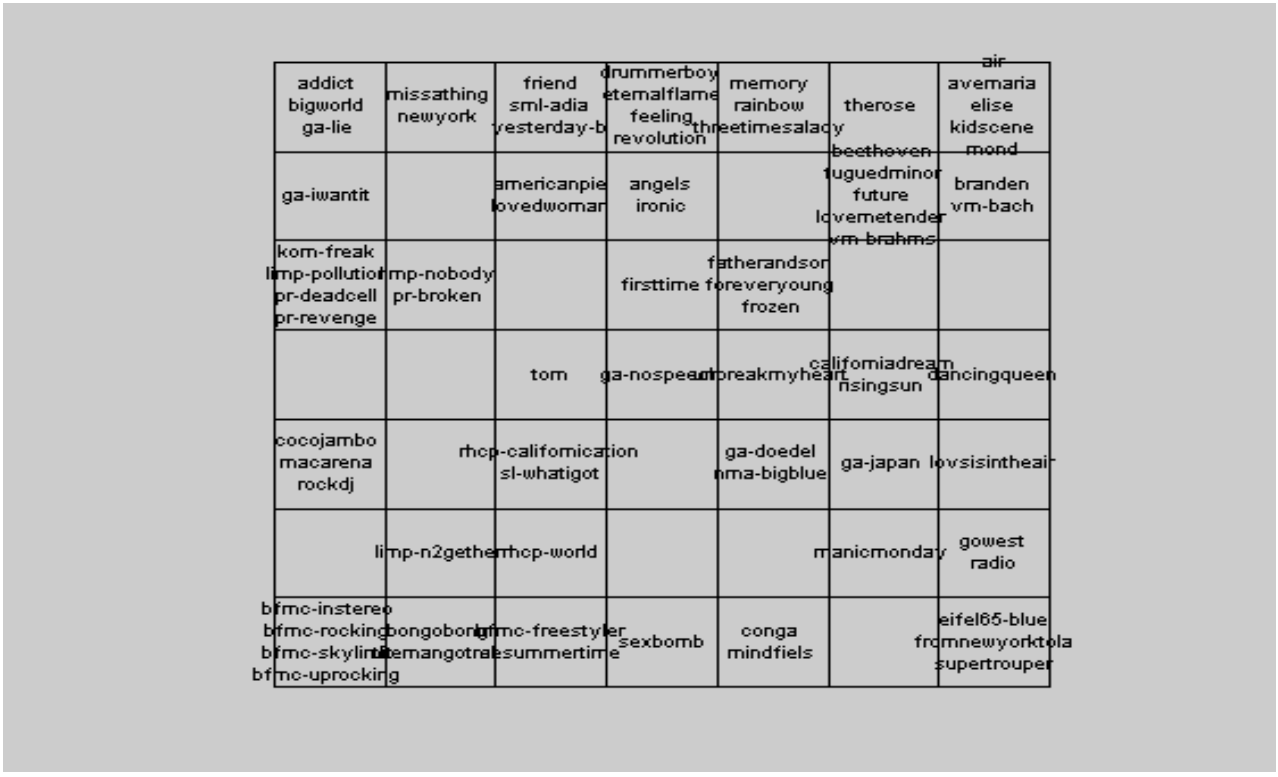

Figure 22. SDH Results after applying SOM algorithm on a dataset of 77 songs from various genres

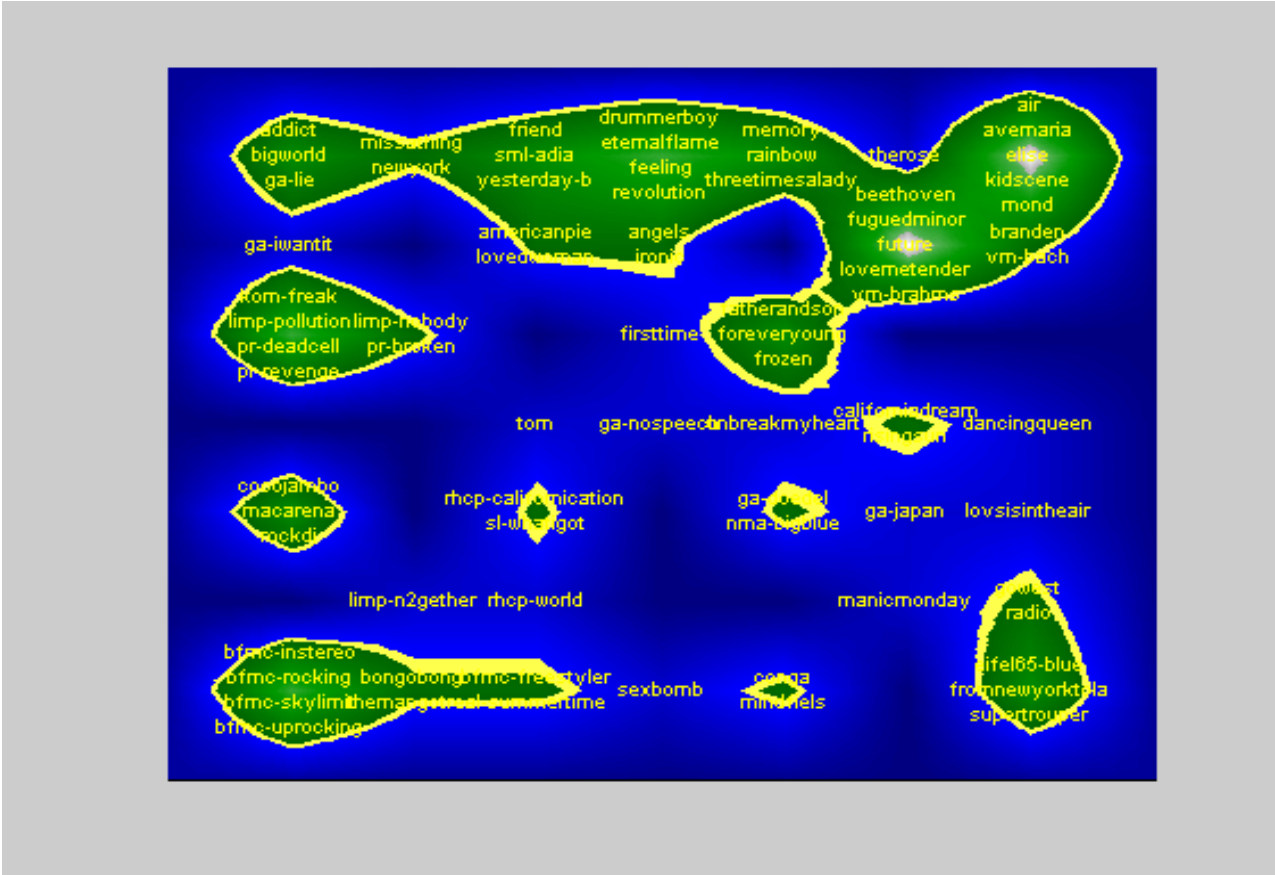

Figure 23. Final Island of Music for the test database of 77 songs

The density function is estimated using the technique presented in the context of the k-means algorithm. Each piece of music votes for the clusters (map units), which represents it best. The first closest model vector of a corresponding unit gets n points, the second n - 1, the third n - 2, and so forth. Thus units, which are close to several pieces of music, will get many points. While clusters, which are not close to any pieces, will hardly have any points. A map unit which is close to many pieces of music is very likely to be close to the center of a cluster, whereas a unit, which does not represent any pieces of music well, is likely to be an intermediate unit between clusters.

Usually one could expect that the second best matching unit is located right next to the first best matching. Thus dividing the hit response of a data item 2/3 for the best matching and 1/3 for the second best matching would lead to the same results with some fuzziness. The often non-spherical shape of the clusters (islands) becomes more apparent and seemingly separated clusters might get connected.

Note that it is not always the case that the first and second best matching units lay next to each other. In fact some quality measures to compare trained SOMs have been developed upon these criteria. However, it is rather unlikely that the two units are separated completely on the map and mostly they will both be located in the same map area. Using $n = 3$ connects more islands and lets them grow bigger. For $n = 4$ the differences to $n = 3$ are not very obvious, however, the islands are slightly more connected and the higher n gets, the more islands will become connected until finally only one big island remains with its peak around the centre of the map.

This visualization has been implemented using interpolating units. Between each row and column units have been inserted and around the whole map as well. These interpolating units do not have a

corresponding model vector and are only assigned interpolated values of the approximated density function.

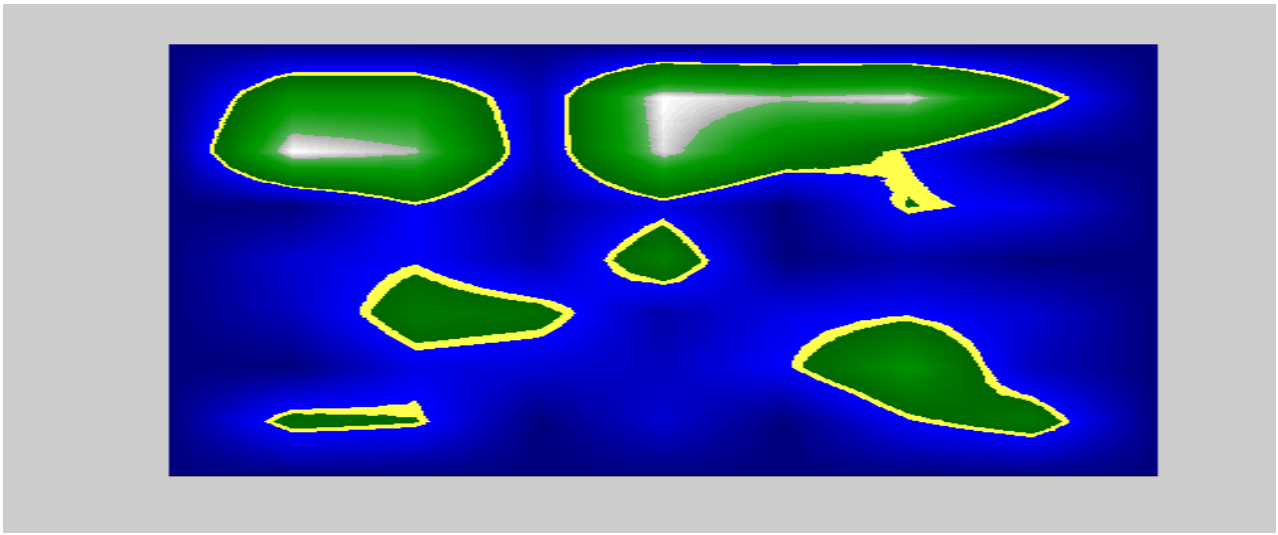

Figure 24. Island of Map with Voting N=3

The units inserted around the map are inserted so that the islands are bounded by the map area. The values of these boarder units are set to 1/10 of their immediate neighbor within the map, and thus are always under water. All density and interpolated density values are then interpolated by Matlab using the *pcolor* function in combination with shading *interp* to create the islands of Figure above. If desired it would be possible to use fractal algorithms or textures to create more naturally looking islands.

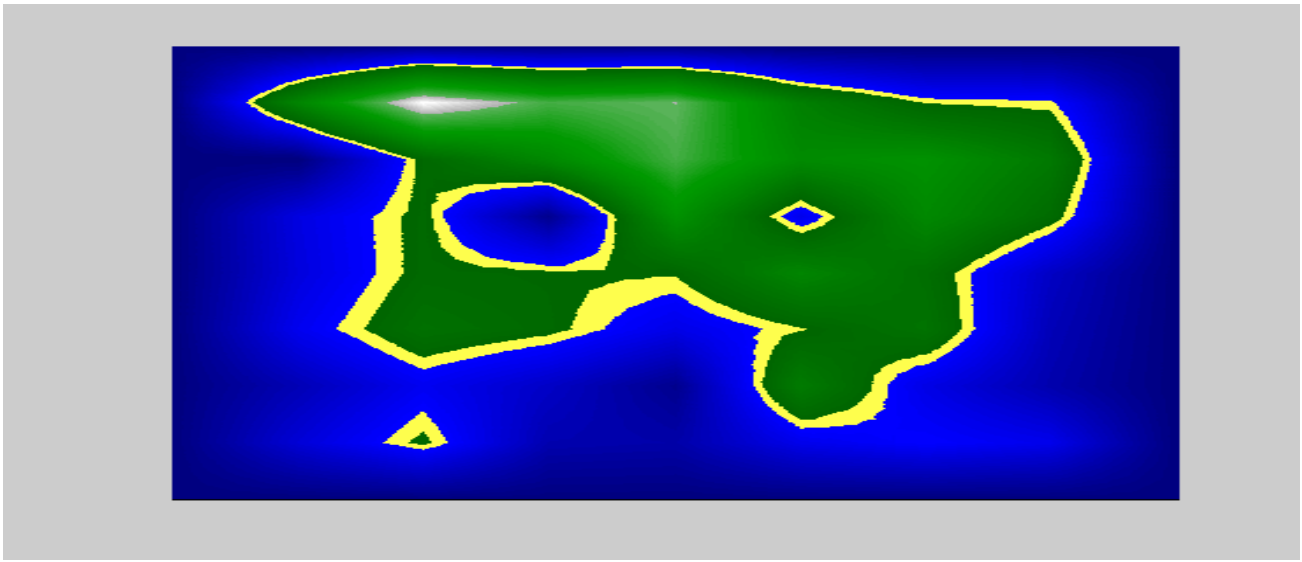

Figure 25. Island of Map with Voting N=10

CHAPTER 4
Conclusions and Future Extensions

4.1 Conclusions of Genre Classifications

The obtained classification accuracy of 86.77 % and 83.11 % is encouraging and comparable to other state-of-the-art algorithms evaluated at the MIREX contest (annual Audio Processing algorithm contest). By analyzing carefully confusion matrices, one can notice that classification errors make sense: for example, on the MAGNATUNE dataset, 20.59 % of Punk excerpts are classified as Rock. There is indeed a clear overlap between these two genres and the misclassified examples may have been probably better described as belonging to both classes.

From the above confusion matrix it is also evident that Log-Compressed energies give better representation of the audio signal as compared to MFCC's feature vectors. Using Log-Compressed energy we were able to achieve 86.77 % of accuracy as compared to 83.11 % of classification using MFCC's as feature vectors. Overall we suggest using both feature vectors for audio genre classification task, as MFCC's give a representation of musical timbre which is equally important as Log-compressed energies which gives better representation of perception model for audio files.

Overall, we find that research is evolving from purely objective machine calculations to techniques where learning phases, training data sets, and preliminary knowledge strongly influence performance and results. This is particularly comprehensible for music genre classification, which has always been influenced by experience, background and sometimes personal feeling. But even in several other classification domains, music related or not, many outstanding solutions exist where machine learning plays a fundamental role, complementary to signal processing.

4.2 Conclusions of Chapter 3

This final chapter summarizes the work presented in this thesis. In addition, opportunities for future research are pointed out and briefly discussed. In this thesis, models and techniques from the fields of signal processing, psychoacoustics, image processing, and data mining were combined to develop a system which automatically builds a graphical user interface to music archives, given only the raw music collection with no further information, such as the genres to which the pieces of music belong.

The most challenging part is to compute the perceived similarity of two pieces of music. Even though currently no final solution to this can be offered, a novel and straightforward approach based on psychoacoustic models is presented and evaluated using a collection of 77 pieces of music. Despite being far from perfect, this approach yields encouraging results.

A neural network algorithm, namely the self-organizing map, is applied to organize the pieces of music so that similar pieces are located close together on a 2-dimensional map display. A novel visualization technique is applied to obtain the map of islands, where the islands represent clusters in the data. To support navigation in unknown music collections, methods to label landmarks, such as mountains or hills, with descriptions of the rhythmic and other properties of the music located in the respective area, are presented. The Islands of Music have not yet reached a level, which would suggest their commercial usage; however, they demonstrate the possibility of such a system and serve well as a tool for further research. Any part of the system can be modified or replaced and the resulting effects can easily be evaluated using the graphical user interface.

4.3 Future Work

Much research is being conducted in the area of content-based music analysis with new results being published frequently. Incorporating these results into the presented system would increase the quality of the Islands of Music.

Based on the approach presented in this work there are some immediate possibilities that might yield improvements. One major problem is the loudness of the pieces of music. Most pieces in the presented collection are from different sources with significant differences regarding the recorded loudness. The loudness has direct impact on the perceived beats and thus strong influence on the whole system. Methods to normalize the loudness would increase the quality.

Another interesting aspect is the sequencing. Each piece of music is divided into 6-second sequences and only a small subset of these sequences is further analyzed to reduce the computational load. Using more sequences or even overlapping them would result in more accurate representations of the pieces of music. Furthermore, the optimal length of the sequences is not yet decided. When calculating the amplitude modulation less then half of the obtained FFT coefficients are used, in particular those which correspond to the modulation frequencies below 10Hz. Thus it would be possible to reduce the length of the sequences to 3 seconds without modifying any other parts of the system. The advantage of a shorter sequence is that it is less likely that it contains more than one specific style.

The applied image processing filters, which emphasize important aspects in the fluctuation strength images, need to be evaluated more thoroughly as well. Alternative parameter settings as well as alternative filters should be considered. In this report several methods to represent a piece of music based on the representation of its sequences were discussed. The finally chosen method, the median, does not coincide with intuitive assumptions; however the alternatives presented were not able to produce significantly better results. A thorough analysis is necessary and perhaps a method, which combines the advantages of the median with the advantages of the other methods presented, could be developed.

Depending on the dataset alternative ways to label the mountains and hills on the islands could be developed which better help in understanding the genres they represent. It is unlikely that such improved labels can be derived directly from the modified fluctuation strength (MFS) data thus the incorporation of different content-based approaches to analyze music is desirable.

4.4 Applications

In Chapter 2 and Chapter 3 we have discussed algorithms and techniques that can be used to better organize our existing music databases. In future we see these algorithms being used by existing service providers, for e.g. Apple ipod. We below present a dummy interface of how in future we can see ipod improving upon the way user can navigate through their music collections.

Future extension of Apple's ipod

Figure 26. On the left is shown the present interface of ipod, while on right we present a dummy future interface of ipod using our interface.

Chapter 5
References

[1] R. Dannenberg, J. Foote, G. Tzanetakis, and C. Weare, "Panel: new directions in music information retrieval," *Proc. Int. Computer Music Conf.*, Habana, Cuba, Sept. 2001.
[2] F. Pachet and D. Cazaly, "A taxonomy of musical genres," *Proc. Content-Based Multimedia Information Access (RIAO)*, Paris, France, 2000.
[3] F. Pachet, J.J. Aucouturier, A. La Burthe, A. Zils, and A. Beurive, "The cuidado music browser: an end-to-end electronic music distribution system," *Multimedia Tools Applicat.*, 2004, Special Issue on the CBMI03 Conference, Rennes, France, 2003.
[4] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," CUIDADO I.S.T. Project Rep., 2004.
[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 5, pp. 293–302, July 2002.
[6] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," *Proc. 6th Int. Symp. Music Information Retrieval*, London, UK, 2005, pp. 604–609.
[7] K. West and S. Cox, "Finding an optimal segmentationfor audio genre classification," *Proc. 6th Int. Symp. Music Information Retrieval*, London, UK, 2005, pp. 680–685.
[8] N. Scaringella and G. Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals," *Proc. 6th Int. Symp. Music Information Retrieval*, London, UK, 2005, pp. 666–671.
[9] E. Gomez, A. Klapuri, and B. Meudic, "Melody description and extraction in the context of music content processing," *J. New Music Res.* vol. 32 no. 1, 2003.
[10] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003
[11] A. Berenzweig, D. Ellis, and S. Lawrence, "Using voice segments to improve artist classification of music," *Proc. AES 22nd Int. Conf.Virtual, Synthetic Entertainment Audio*, 2002.
[12] M. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," *Proc. 6th Int. Symp. Music Information Retrieval*, London, UK, 2005, pp. 594–599.
[13] H. Soltau, T. Schultz, M. Westphal, and A. Waibel, "Recognition of music types," *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing (ICASSP)*, Seattle, WA, USA, 1998, vol. II, pp. 1137–1140
[14] N. Casagrande, D. Eck, and B. Kegl, "Geometry in sound: a speech/music audio classifier inspired by an image classifier," *Proc. Int. Computer Music Conf. (ICMC)*, 2005.
[15] A. Flexer, E. Pampalk, and G. Widmer, "Novelty detection based on spectral similarity of songs," *Proc. 6th Int. Symp. Music Information Retrieval*, London, UK, 2005, pp. 260–263.
[16] K. West, S. Cox, "Features and classifier for the automatic classification of musical audio signals", *Proc. Of the 5th Int. Conf. on Music Information Retrieval (ISMIR), Barcelona, Spain*, 2004.
[17] G. Tzanetakis, P. Cook, "Musical genre classification of audio signals", *IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 5*, July 2002.
[18] N. Scaringella, G. Zoia, "On the modelling of time-information for automatic genre recognition systems in audio signals", *Proc. of the 6th Int. Conf, on Music Information Retrieval, London, UK*, 2005.
[19] L. Rabiner, B.H. Juang, *Fundamentals of speech recognition*, Englewood Cliffs, NJ, Prentice-Hall, 1993.
[20] J.J. Aucouturier, F. Pachet, "Improving timbre similarity: how high's the sky?", *Journal of Negative Results in Speech and Audio Sciences*, 2004.
[21] http://www.music-ir.org/
[22] http://www.magnatune.com
[23] F. Nack and A. Lindsay. Everything you wanted to know about MPEG7 - Part 1. *IEEE MultiMedia*, pages 65–77, July/September 1999.
[24] G. Peeters, S. McAdams, and P. Herrera. Instrument Sound Description in the Context of MPEG-7. In *Proceedings of the ICMC2000*, 2000.
[25] A. Rauber and D. Merkl. The SOMLib Digital Library System. In *Proceedings of the 3rd Europ. Conf. On Research and Advanced Technology for Digital Libraries (ECDL'99)*, Paris, France, 1999. Lecture Notes in Computer Science (LNCS 1696), Springer.
[26] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. WEBSOM-self organizing maps of document collections. *Neurocomputing*, 21:101–117, 1998.
[27] J. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999.

[28] M. Liu and C. Wan. A Study of 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.
[29] J. T. Foote. Content-based retrieval of music and audio. In C. Kuo, editor, *Proceedings of SPIE Multimedia Storage and Archiving Systems II*, volume 3229, pages 138–147, 1997.
[30] B. Logan. Mel Frequency Cepstral Coefficients for Music Modelling. In *International Symposium on Music Information Retrieval (MUSIC IR 2000)*, Plymouth, Massachusetts, 2000.
[31] E. D. Scheirer. *Music-Listening Systems*. PhD thesis, MIT Media Laboratory, 2000.
[32] M. Fruhwirth and A. Rauber. Self-Organizing Maps for Content-Based Music Clustering. In *Proceedings of the 12th Italien Workshop on Neural Nets (WIRN01)*, Vietri sul Mare, Italy, 2001. Springer.
[33] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.
[34] E. Zwicker and H. Fastl. *Psychoacoustics, Facts and Models*, volume 22 of *Springer Series of Information Sciences*. Springer, Berlin, 2nd updated edition, 1999.
[35] H. Fastl. Fluctuation strength and temporal masking patterns of amplitude-modulated broad-band noise. *Hearing Research*, 8:59–69, 1982.
[36] E. Zwicker, G. Flottorp, and S. S. Stevens. Critical band width in loudness summation. *Journal of the Acoustical Society of America*, 29:548–557, 1957.
[37] E. O. Brigham. *The Fast Fourier Transform*. Prentice Hall, Englewood Cliffs, NJ, 1974.
[38] R. Bladon. Modeling the judgment of vowel quality differences. *Journal of the Acoustical Society of America*, 69:1414–1422, 1981.
[39] S. Weil. Music Analysis and Characterization. Student Project (CS 152 - Neural Networks), Harvey Mudd College, Claremont, CA. http://www.newdream.net/~sage/nn, 1999.
[40] D. P. W. Ellis. *Prediction-Driven Computational Auditory Scene Analysis*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, 1996.
[41] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review.*ACM Computing Surveys*, 31(3):264–323, 1999.
[42] T. Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
[43] C. M. Bishop, M. Svens´en, and C. K. I. Williams. GTM: A principled alternative to the Self-Organizing Map. *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, volume 1112 of *Lecture Notes in Computer Science*, pages 165–170, Berlin, 1996. Springer.
[44] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.
[45] T. Kohonen. New developments of learning vector quantization and the self-organizing map. In *SYNAPSE'92, Symposium on Neural Networks*, Osaka, Japan, 1992. Alliances and Perspectives in Senri.
[46] D. Hearn and M. P. Baker. *Computer Graphics, C Version*. Prentice Hall, NJ, 2nd edition, 1997.