TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

M A G I S T E R A R B E I T

Discrimination and Retrieval of Environmental Sounds

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

unter Anleitung von
ao. Univ. Prof. Mag. Dr. Horst Eidenberger

durch

Dalibor Mitrovic
Matr. Nr. 9925385

Wipplingerstraße 32
A-1010 Wien

Wien am 7. Dezember 2005

Datum

Unterschrift

Abstract

The human auditory sense may be regarded as the second most important sense after the sense of sight. This valuation is reflected in the field of information retrieval where until recently research concentrated on visual information retrieval. Even research in audio retrieval (AR) focused on one single aspect of hearing, namely understanding of speech. With the upcoming of large music databases in recent years, a second area of AR gained importance: music information retrieval (MIR). The goal of MIR is to enable efficient search and retrieval in the music databases mentioned above. The latest research area in the domain of audio retrieval is the retrieval of environmental sounds. One may argue that environmental sound retrieval deserves a more prominent role than it has. Most sounds humans hear are neither speech nor music but various environmental sounds. By incorporating environmental sounds into retrieval systems, a vast amount of additional information becomes available.

In this thesis the applicability of a range of audio features in the domain of environmental sound retrieval is investigated. Furthermore state-of-the-art techniques in audio retrieval are identified by a broad survey of relevant literature covering all three areas of AR (speech, music, and environmental sounds). The quality of the features is examined with three different classification techniques. Finally, a set of novel audio features, developed by the author, is compared to established features. Results indicate that further research is necessary. There is particularly a lack of low-dimensional and computationally cheap audio descriptors suitable for the use in environmental sound retrieval.

Zusammenfassung

Der menschliche Hörsinn kann als der zweitwichtigste Sinn nach dem Sehsinn betrachtet werden. Diese Wertung spiegelt sich im Information Retrieval wider; bis vor kurzem konzentrierte sich die Forschung dort auf visuelle Information. Selbst das Audio Retrieval war nur auf einen Teil des menschlichen Hörens fokusiert und zwar auf die Spracherkennung. Mit dem Aufkommen von großen Musikdatenbanken gewann ein zweites Forschungsgebiet, Music Information Retrieval, an Gewicht. Das Ziel des MIR is es, effizientes Suchen und Finden in den oben genannten Musikdatenbanken zu ermöglichen. Das jüngste Forschungsgebiet im Audio Retrieval stellt das Erkennen von Umgebungsgeräuschen dar. Erkennung von Umgebungsgeräuschen verdient eine wichtigere Rolle als bisher: schließlich hört der Mensch meist Umgebungsgeräusche und nicht Sprache oder Musik. Durch das Miteinbeziehen von Umgebungsgeräschen in das Retrieval erschließt sich eine Fülle an zusätzlicher Information.

In dieser Arbeit werden in einer breit angelegte Literaturrecherche aktuelle Methoden aus allen drei Teilbereichen des Audio Retrievals identifiziert. Weiters wird die Eignung weit verbreiteter Audiofeatures für das Retrieval von Umgebungsgeräuschen untersucht. Die Qualität der Features wird mit drei Klassifikationsmethoden beurteilt. Außerdem wird ein neues Audiofeature eingeführt und mit etablierten verglichen. Die Ergebnisse verdeutlichen, dass weitere Forschungarbeit auf dem Gebiet der Umgebungsgeräuscherkennung notwendig ist. Insbesondere ist ein Mangel an niedrigdimensionalen und leicht zu berechnenden Audiofeatures festzustellen, die für dieses spezielle Einsatzgebiet geeignet sind.

Contents

Acknowledgments

I extend my sincere gratitude and appreciation to many people who made this masters thesis possible.
Especially I am grateful to my parents whose support enabled me to achieve my goals.
Special thanks are due to Ms. Doris Divotkey and Ms. Karyn Laudisi. Furthermore, I want to thank my supervisors ao. Univ. Prof. Mag. Dr. Horst Eidenberger and Univ. Prof. Dipl. Ing. Dr. Christian Breiteneder.

This work has received financial support from the Austrian Science Fund (FWF). It is part of the audio subproject of the VizIR project (grant no. P16111-N05) and is a result of the joint scientific work of Dalibor Mitrovic and Matthias Zeppelzauer on audio retrieval [45], [66].

1 Introduction

Multimedia information retrieval is a growing research field that gained importance in recent years, due to the increasing number of available digital media. Traditionally, research focuses on visual information retrieval (VIR). The rise of audio information retrieval was motivated by the development of efficient audio compression techniques that support the distribution of digital audio. Another application of audio retrieval is multimodal information retrieval, where visual, textual, and acoustic information is combined to take advantage of synergetic effects. Audio recognition is also employed for automatic extraction of semantic annotations in multimedia databases.

This thesis addresses the retrieval of environmental sounds. Therefore a broad set of audio features and several classifiers are surveyed. Additionally, a new set of features is introduced and their quality for a selected set of classes of environmental sounds is evaluated. Due to the complexity of the retrieval task, the quality of non-speech sound recognition is typically lower than the quality of speech recognition, which is already well understood. Retrieval results presented in this work are comparable to the results of state-of-the-art research in the area of environmental sound recognition.

1.1 Motivation and Problem Statement

Audio recognition and retrieval has been an important and challenging research field for more than fifteen years. Although the research community yielded great technical advances in the past, work in this area is still at a preliminary stage. The long-term goal is to achieve results comparable to the human sense of hearing. The human auditory sense provides optimal performance, since it is able to bridge the semantic gap described in Subsection 2.2. Audio recognition and retrieval techniques can at best narrow the semantic gap. Although there is a huge research community, publishing a vast amount of scientific papers for many years, there are still a lot of unsolved problems. The representation of audio signals by numerical features is currently at a low level of abstraction that does not consider semantic information. Measuring similarity of audio signals is a very difficult task, still open to research. Audio retrieval is currently only applicable to a limited

domain of sounds. In contrast to speech recognition, the domain of environmental sounds is nearly infinite. The retrieval quality decreases rapidly with an increasing number of classes that have to be distinguished. Besides, the quality of retrieval degrades with increasing inhomogeneity of the audio samples that belong to the same class. Furthermore, the partitioning of sounds into disjoint classes is ambiguous and subjective due to cultural influences. Another challenge is the representation of queries in retrieval systems. Early approaches employed query-by-example techniques. Later, query-by-humming gained importance particularly in the field of music retrieval [22]. A retrieval task always is a tradeoff between universality and assumptions - about the domain, about the media, and about the user. The retrieval quality is proportional to the number of assumptions the investigation is based on. Most investigations examine a limited domain of sounds. For example this thesis deals with animal sounds and does not consider arbitrary environmental sounds. Other assumptions concern knowledge about the media objects. For example whether the media objects contain sounds from different classes or not. Furthermore, knowledge about the user of the retrieval system may lead to new assumptions. For example, the users' expectations to the retrieval system and the purpose the system is used for. All assumptions together allow for optimization of the retrieval task.

The problem of content-based audio retrieval can be stated as follows: Content-based audio retrieval concerns itself with searching in multimedia databases for audio samples specified by a query that describes properties of the desired audio samples. Often retrieval is the task of deriving a parametric model from raw data. From a given set of audio signals, each annotated with a class label, a more compact abstract numerical representation by features must be derived that characterizes the properties of the classes well. During the training phase a (parametric) model, the classifier, is fit to the feature-data. The goal of training is to correctly predict the class membership of all possible audio signals in the scope of the defined classes. Based on the parametric model, retrieval is performed by defining and evaluating a query.

1.2 Contribution

Distinction of different classes of environmental sounds is one goal of this thesis. The author's contribution to this research field is represented by a thorough investigation of the applicability of state-of-the-art audio features in the domain of environmental sound recognition. Therefore a database consisting of several hundred environmental sounds was built. Traditional features developed for speech recognition and features applied in audio segmentation and music retrieval are considered. Additionally, a set of novel features is introduced and compared with established audio features. The new features are time-based and follow an intuitive approach to describe the waveform of a signal. The quality of the features investigated is evaluated by a representative set of popular classifiers. Furthermore, an extensive survey of state-of-the-art features and classifiers is given. Finally, a comprehensive overview of related research in the field of content-based audio retrieval is provided.

1.3 Applications

Environmental sound retrieval has a wide range of applications. It may play an important role in applications for handicapped people. Such a technique could be part of a supporting system for the deaf, providing information about the surrounding environment. A deaf person could be equipped with a microphone and a mobile device that is responsible for retrieval. The user would be visually informed by the application about interesting or dangerous events, indicated by sounds.

A popular application is automatic surveillance. It usually employs multiple cameras and microphones to monitor an area of interest. Such a system produces huge amounts of data that contain only little information. Environmental sound retrieval may be applied, for example to detect approaching cars that can not be seen by the mounted cameras.

A traditional research field is the annotation of time-dependent media. Environmental sound retrieval may be part of a system that automatically generates meta-information from audio and video streams. A related ap-

plication is the annotation of movies in a multimedia database to improve search capabilities.

Additionally, life logging applications could take advantage of such a technique. Life logging applications accompany human users during their working life and leisure time and automatically capture and annotate events of interest in a multimodal diary. Usually, life logging applications employ multiple different sensors, such as video cameras, microphones, GPS, accelerometers, and thermometers [1]. Information is extracted from the single signal and combined with retrieved data of other measured signals. The resulting diary consists of the retrieved annotations combined with a time stamp. A thorough survey of applications related to content-based audio retrieval and environmental sound retrieval is given in Section 5.

1.4 Organization

The remainder of this thesis is organized as follows: In Section 2, the principles of pattern recognition, retrieval and digital audio are given. Section 3 addresses the experiments and discusses features and classifiers. Results are depicted in Section 4. A survey of related work is performed in Section 5. Finally, in Section 6 conclusions and future work are presented.

2 Background

In this section some basic ideas of audio retrieval are discussed. First, the field of information retrieval is surveyed. Then, the author presents the fundamentals of pattern recognition. Finally, basics of digital audio are discussed.

2.1 Information Retrieval

Information retrieval is concerned with searching documents in a database by a textual query. Early applications focused on retrieval of text documents. Information retrieval is performed by searching in the documents themselves or by searching for documents by annotated metadata. A popular application of information retrieval are search engines in the World Wide Web. Pioneers in the area of information retrieval are Salton [52] and van Rijsbergen [60].

In the last decades the number of available media has grown. Audio and video data have become available due to the development of efficient compression techniques and the distribution of multimedia over the internet. Traditional text-based information retrieval is not appropriate for retrieval of audio and video data. Manual creation of textual metadata from multimedia objects by humans is not applicable because it is too time-consuming, error-prone and costly. The limitations of metadata-based retrieval techniques can be overcome by examining the content of media objects. Content-based information retrieval is a separate branch of research of information retrieval where information about audio and video documents is extracted directly from their content. There is no need for a priori knowledge concerning the documents. Depending on the media type, we distinguish between content-based image retrieval (CBIR), content-based video retrieval (CBVR) and content-based audio retrieval (CBAR).

2.2 Pattern Recognition

Most approaches dealing directly with the content of multimedia documents are applications of pattern recognition. Pattern recognition is concerned

with analyzing and classifying data objects by contained patterns. A pattern recognition system consists of multiple parts. A sensor (e.g. a microphone or video camera) provides the system with the raw signal data. The size of the data is reduced by feature extraction. This results in a more abstract description that represents the most meaningful information that best characterizes the signal. Based on this representation, classification is performed. Classification is a process that groups similar patterns represented by features together. Figure 1 illustrates this process. In a content-based retrieval

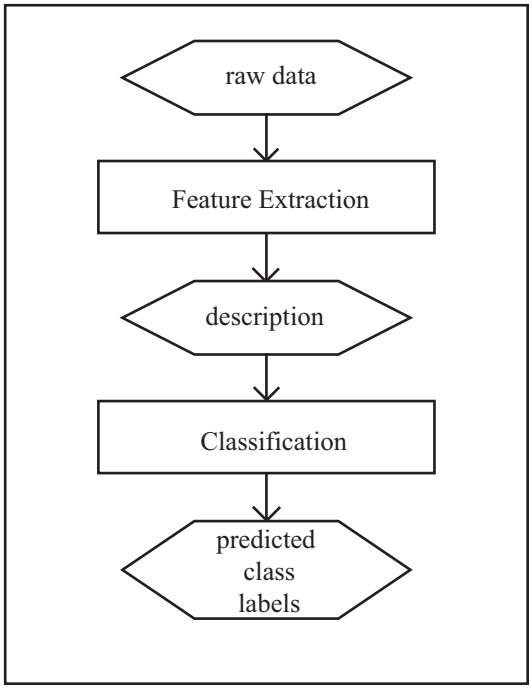


Figure 1: Sequencediagram of a typical pattern recognition task.

application the user addresses queries to the retrieval system. Queries can be expressed in different ways. One approach is query-by-example, where the query is of the same media type as the documents in the database. Alternatively, a textual description of the favored document (e.g. "find sounds of cars" or "find pictures of cats") can be formulated as the query.

According to Watanabe, patterns are "the opposite of chaos" [62]. A pattern has a structure that is characterized by features, which are numerical representations of that pattern, such as the height of a person or the pitch

of a human voice. A feature is regarded as a mapping from pattern space (raw data) to feature space. The value of a feature is usually represented by a scalar. In practice, several features are combined into a feature vector.

Feature extraction denotes the process of computing features. In context of content-based retrieval, features often represent the coefficients of basic signal processing transforms such as the Fast Fourier Transform (FFT) or the Discrete Cosine Transform (DCT). The advantage of such transforms is that a few coefficients suffice to represent most of the original signal. Due to this property, these transforms are applied in signal compression techniques such as JPEG and MPEG. Subsection 3.4 gives a thorough discussion of a variety of audio features. This thesis describes the application of the features in content-based audio retrieval.

As mentioned above, features are often combined to feature vectors. Feature selection is the process of choosing a maximal informative subset from a given set of features. Statistical methods, such as the Principal Component Analysis (PCA) that maximizes the variance among the features, are often applied for feature selection. Besides, PCA can be used to generate new features based on existing ones [39].

The objective of classification is to predict the class membership of a pattern represented by a feature vector. A class ω_i is defined by a class label $i \in N$. Each pattern/feature vector belongs to exactly one class. A classifier can be regarded as a function $c(\mathbf{x})$ of a feature vector $\mathbf{x}$ with:

$$c(\mathbf{x}) = i \Leftrightarrow \mathbf{x} \in \omega_i \tag{1}$$

The outputs of a classifier are the predicted class labels of the feature vectors. Most classifiers have to be trained before they can be applied to arbitrary test patterns. For this purpose, the sample database is split into training and test sets. The training samples are chosen randomly. The training set usually is much smaller than the test set. During training, the classifier determines the class boundaries based on feature vectors from the training set. After training, the classifier is fit to the data and ready for classification. The quality of classification is evaluated using a test set. The test set contains feature vectors that are not contained in the training set. A classifier should be able to correctly classify not only the training vectors,

but all arbitrary vectors belonging to one of the selected classes. This is called the generalization ability of a classifier [15]. In Subsection 3.5, three classifiers employed in the investigations are presented in detail.

The quality of content-based retrieval depends on the features that represent the signal and on the classifiers that discriminate between classes of signals. An optimal feature shows minimal variations inside a class and high variations between multiple classes. A good representation of data by features is necessary for successful classification. Results of the classifiers basically depend on the quality of the features. No feature is a priori good or bad. The quality of a feature has to be analyzed in context of the input data, the application domain, and the classes that are distinguished. Similarly, classifiers cannot be evaluated in isolation. They have to be considered together with the features on which they operate.

Pattern recognition tasks (e.g. remote sensing, computer vision, image understanding, and content-based retrieval) are inversions of well-posed problems. Computer graphics is the well-posed inversion of pattern recognition and content-based image retrieval. Similarly, sound synthesis is the well-posed inversion of content-based audio retrieval. In general an inverse problem concerns with the estimation of model parameters by manipulation of observed data [63]. The inversion of a well-posed problem is often ill-posed. The term ill-posed means that the conditions mandatory for well-posed problems are not met. Conditions for well-posed problems are defined by Hadamard in [25]. According to Hadamard, a well-posed problem has the following properties:

1. A solution exists,
2. the solution is unique, and
3. the solution depends continuously on the data in some reasonable topology.

Content-based retrieval is an ill-posed problem. In a retrieval task, model parameters are derived from input data (audio, image or video data). Model parameters are terms, properties and concepts that may represent class la-

bels (e.g. terms as “car” and “cat,” properties as “male” and “female,” and concepts as “outdoor” and “indoor”).

The semantic gap is related to the ill-posed nature of content-based retrieval. The semantic gap refers to the mismatch between high-level concepts and low-level descriptions. In content-based retrieval the semantic gap is positioned between the content of media and textual information describing the semantics of the content. The gap cannot be bridged due to the ill-posed nature of content-based retrieval. Today the goal of the research community is to narrow the semantic gap as far as possible. All content-based retrieval branches, such as CBAR, CBVR, and CBIR suffer from the problems introduced by the semantic gap and apply similar techniques to narrow it.

2.3 Content-Based Retrieval Systems

CBIR came up in the 1990s. One of the first image retrieval systems was QBIC [19]. The QBIC system is able to query a multimedia database by example images or videos. Around the same time the first investigations on CBAR were performed. Pioneering work is presented by Wold, Blum, and Wheaton in [64]. The authors developed an audio retrieval system called Muscle Fish that is able to distinguish a wide range of sounds.

Multimedia retrieval systems have a complex architecture. The core of a retrieval system is the database that stores the (multimedia) documents. Additionally, it stores annotated metadata and extracted features. Features are automatically computed by a feature extraction mechanism. Traditionally, annotations were created manually by human users. Modern systems support automatic extraction of annotations. A search engine is connected to the database that receives queries from the user. A retrieval system may support multiple types of queries. Query-by-example techniques directly use documents as query objects. The retrieval system computes features from the query documents and the search engine tries to locate similar documents in the database by applying a similarity model. For example, in the vector space model, similarity can be measured by a distance measure such as the Euclidian distance. Alternatively the classifier itself may be employed to es-

timate similarity. Another method is query-by-text, where the user defines the desired class of documents or terms describing the documents. Query-by-text makes use of media annotations stored in the database. Another part of the retrieval system is responsible for visualization of the retrieved documents. It provides the user with an interface to browse the returned media objects.

Different evaluation methods for retrieval systems do exist. The most popular measures are Recall and Precision. Recall is the proportion of retrieved relevant documents of all relevant documents in the database. Let *Ret* be the set of retrieved documents and *Rel* the set of relevant documents in the database. Recall R is defined as:

$$R = \frac{|\{Ret \cap Rel\}|}{|\{Rel\}|} \tag{2}$$

Precision is the percentage of relevant documents retrieved in relation to the total number of documents retrieved. Precision P is defined as:

$$P = \frac{|\{Ret \cap Rel\}|}{|\{Ret\}|} \tag{3}$$

Recall and Precision are inversely related. Precision decreases with increasing Recall and vice versa. The tradeoff between Recall and Precision is usually illustrated in a Recall-Precision Graph. The Recall-Precision Graph shows the Precision on the ordinate for different Recalls on the abscissa. The Recall-Precision pairs are obtained by varying the number of retrieved documents. A typical Recall-Precision Graph is given in Figure 2.

2.4 Content-Based Audio Retrieval

The rising number of audio, video, and image databases brings forth the need for efficient retrieval. The exponential growth of computational power enables multiple applications for content-based retrieval, such as real-time surveillance, video analysis, and music information retrieval. These trends encourage research in this area. Today several hundred scientific publications are published every year.

CBAR is a relatively young research area. The techniques applied are tightly coupled to CBIR. CBAR additionally employs methods of speech

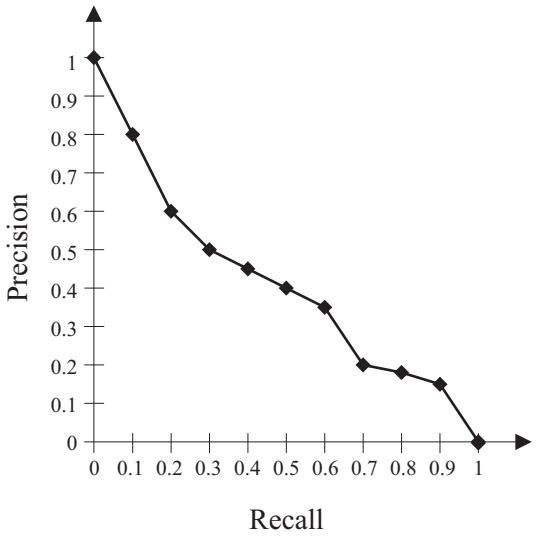


Figure 2: A typical Recall-Precision Graph, illustrating the tradeoff between Recall and Precision.

recognition. Speech recognition is a research field with long tradition. It was one of the first challenges in digital audio analysis. Due to the similar nature of the approaches in both research areas, knowledge from speech recognition may be reused in CBAR. Today speech recognition is well understood and well engineered. It is extensively surveyed by Rabiner and Juang in [48]. The results of CBAR currently cannot compete with those of speech recognition. The reason for this may be the significant impact of the semantic gap.

There are different branches of research in CBAR. *Segmentation* covers distinction of different types of sound such as speech, music, silence, and environmental sounds. Segmentation is an important preprocessing step used to identify homogeneous parts in an audio stream. Based on segmentation the different audio types are further analyzed by more appropriate techniques such as speech recognition, music information retrieval and environmental sound recognition.

Traditionally speech recognition addresses the recognition of the spoken word on the syntactical level. Besides, research focuses on the recognition of the spoken language. A popular research field is speaker recognition which is employed for authentification. Other investigations deal with the extraction of emotions from human speech.

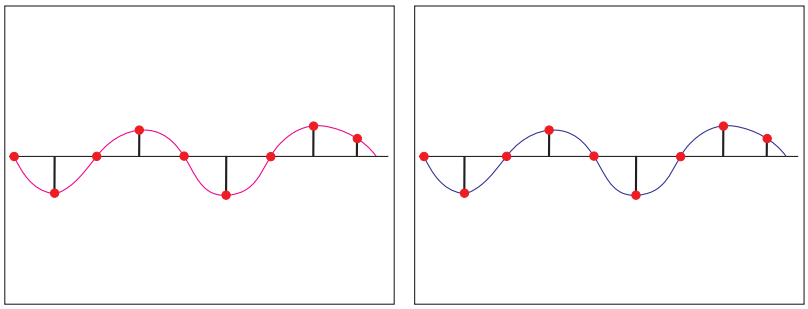
In the last decade analysis and retrieval of music became a popular research field [17]. On the one hand research deals with retrieval of instruments, artists and musical styles. On the other hand researchers concentrate on the extraction of semantic information in pieces of music.

Another research field is environmental sound retrieval which comprises all types of sound that are neither speech nor music. Since the domain of environmental sounds is arbitrary in size, most investigations restrict to a limited domain of sounds. A thorough investigation of related work is given in Section 5.

2.5 Digital Audio

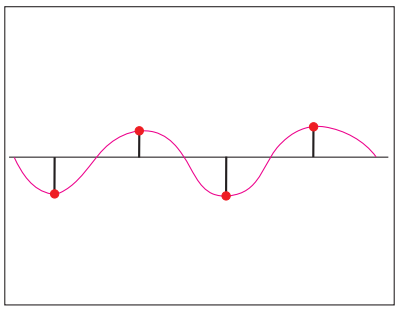
Sound in context of this work is defined as vibrations transmitted through an elastic media (be it solid, aeriform or liquid) that are detectable by the human auditory sense. The detectable vibrations have frequencies ranging from 20 Hz to 20 000 Hz.

Since physical sound is analog it has to be digitized to be processable with digital hardware. Usually digital sound recording means recording a number of samples of that sound at certain time intervals. In order to enable a perfect reconstruction of the digital signal, the analog signal has to be sampled uniformly and at a frequency that is equivalent to at least twice its bandwidth. This theorem is known as the Nyquist-Shannon sampling theorem, illustrated in Figure 3. Pulse Code Modulation (PCM) is a standard technique for digitally encoding analog audio. It dates back to 1937 when a French engineer named Alec Reeves introduced PCM for the purpose of telephone transmission. The analog signal is sampled at uniform intervals and quantized into a digital code. The sampling rate defines the bandwidth of the encoded signal according to Nyquist-Shannon sampling theorem. Besides, the quantization depth is a critical quality measure since it determines the resolution of the amplitude information. Quantization always introduces some noise, known as quantization noise, that is not necessarily audible. A widely known example for digitally encoded analog audio is the CD-Audio standard. It defines a sampling rate of 44 100 Hz and a quantization depth

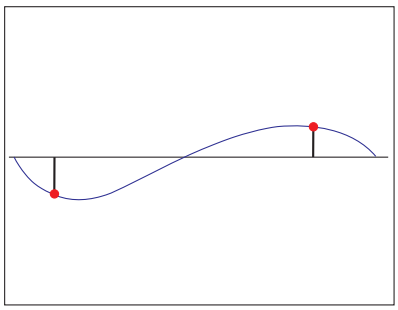


(a) oversampled signal

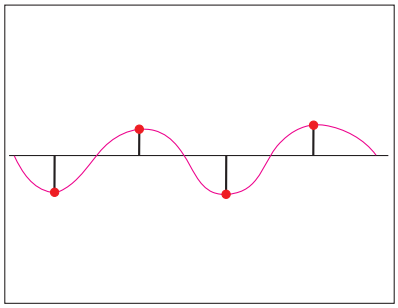
(b) reconstructed oversampled signal



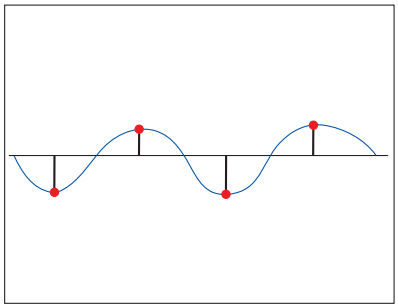
(c) undersampled signal



(d) loss of information in reconstructed undersampled signal



(e) signal sampled with twice its bandwidth (Nyquist-Shannon limit)



(f) reconstructed signal

Figure 3: Audio sampling at different rates: 3(a) and 3(b) illustrate, that no gain is achieved by oversampling. The reconstructed signal in 3(b) is identical to the original signal. 3(c) and 3(d) show the devastating effect of undersampling. The signal cannot be reconstructed properly. Sampling at the optimal rate is depicted in 3(e) and 3(f).

of 16 Bits. Such an encoding preserves all perceivable frequencies and does not introduce audible quantization noise.

3 Experiments

In this section, the performed experiments are described. Different features and classifiers are applied and compared. First, scope and objectives of the experiments are discussed. Then the setup of the experiments and the test environment that supports the experiments are described. Finally, the features and classifiers employed in the tests are presented.

3.1 Scope

Five types of environmental sounds, namely cars, crowds, footsteps, signals, and thunder are chosen for investigation. Sounds of cars, thunder and crowds show significant similarities on the technical and perceptual level. This qualifies the selected classes to measure the quality of features and classifiers without bias.

The goal is to compare techniques in context of the domain of environmental sounds. A system that is able to correctly classify about 80% of the environmental sounds contained in the test set may be regarded as a success. The system learns the differences that characterize the different sounds from a training set that is much smaller than the test set. The techniques applied for retrieval should be easy to compute, to meet the demands of mobile environments such as in a life logging application.

3.2 Setup

The author built a database of sound samples from an internet search. The database contains 557 samples (105 cars, 127 crowds, 118 footsteps, 105 signals and 102 thunder sounds). The data have a sample rate of 11025 Hz, are quantized to 16 bit and are single channel. A sound sample contains one or more repeated sounds of one class. File lengths and loudness levels vary over the samples. Classification is performed on entire sound files (file-based classification). Each sound sample is assigned to exactly one of the five classes.

Numerous experiments are performed to test each feature with each classifier. All experiments have the same structure. An experiment consists of a

number of inputs and outputs with corresponding parameters. The following inputs exist:

- **data** defines the directories where test set and training set are located. Optionally, a file can be specified where the test and training set are stored as a binary file.
- **feature(s)** specifies the feature(s) to compute. One or more features can be defined. Each feature may return a feature vector containing several components. For each feature the corresponding parameters are given.
- **feature selection** defines the components of the feature vectors that are used in the experiment.
- **classifier** denotes the classification technique used and its specific parameters.

Currently, the following outputs are defined:

- **data file** is a binary file where the raw data from the test and training set are stored. The data file includes metadata such as sample rate, file size, file path, class name and class label.
- **feature(s) file(s)** are binary files that store the feature vectors, extracted from the sound samples in test and training set.
- **retrieval evaluation** defines a technique to identify the quality of classification. The current implementation supports Recall and Precision.

The inputs and outputs together with their parameters are stored in an *experiment file*. The uniform structure of experiments enables efficient and consistent tests of various features and classifiers. All experiments are conducted in MATLAB using an extensible framework introduced in Subsection 3.3 that supports experiment files defined as above [44].

The ground truth is common to all experiments. The sample database is split into a test set and a training set. The training set comprises of 12

randomly chosen samples per class, except the training set for the signals class that contains 25 samples because the class members show a wide diversity. This fact is tightly connected to the semantic gap discussed earlier. The training samples are chosen randomly to gain an unbiased training set. The remaining samples form the test set: 93 cars samples, 115 crowds samples, 106 footsteps samples, 81 signals samples and 90 thunder samples. The training set is chosen to be very small (approximately 12.5% of the data) to prove the ability of the classifiers to generalize.

The experiments are split into two test series. In the first run, all features are tested individually. Their quality is evaluated by a set of classifiers. The author employs popular techniques from machine learning and artificial intelligence. These are Learning Vector Quantization (LVQ) and a Support Vector Machine (SVM). Additionally, the K-Nearest Neighbor (K-NN) classifier is applied because of its simplicity. The results of these experiments are discussed in Subsection 4.1. In the second run, the features are combined to improve the quality of retrieval. The corresponding results are illustrated in Subsection 4.3. The large number of experiments enables an objective comparison of the employed classifiers in Subsection 4.5.

3.3 Test Environment

The author implemented an extensible framework that supports the definition of experiment setups by configuration files. Configuration files specify ground truth, test data, features, classifiers, and result output options as discussed in Subsection 3.2. The author decided for the MATLAB environment because it provides a comfortable interface for audio processing and a large number of basic audio algorithms. Furthermore multiple toolboxes exist, such as [47], [5], [16], and [40] that deal with audio analysis, speech recognition and classification.

The goal of the framework is to provide common interfaces for basic pattern recognition tasks such as feature extraction and classification. An experiment represents an entire retrieval process. The framework is able to represent an entire experiment as a short description that may be stored in a configuration file.

The MATLAB framework integrates the implementations of all features employed in the experiments. It encapsulates the feature implementations and provides standardized interfaces for the features. The same functionality is provided for the classifiers. The framework operates on a few data structures that contain the feature data and the raw sample data. Interfaces to features and classifiers operate on these common data structures. Integration of new features and classifiers is performed by implementing an interface that encapsulates its specific logic.

The framework provides a mechanism to store and import sample data and precomputed feature data. This speeds up repeated experiments enormously and allows further analysis of feature data. The structure of the framework is depicted in Figure 4.

The experiments are performed on a PC with an Athlon 64 3000+ and 512 MB of RAM. MATLAB version 6.5 is used for the experiments.

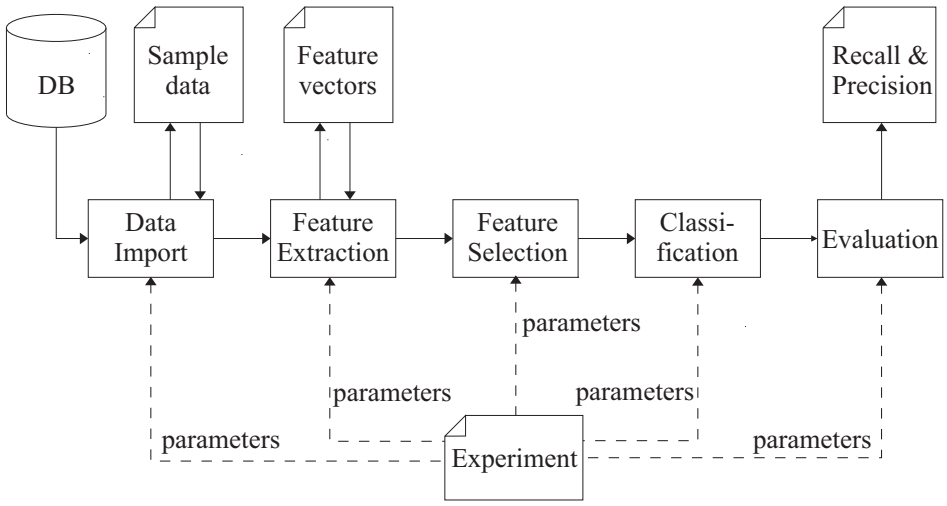


Figure 4: The MATLAB framework employed for the experiments. Each experiment is represented by a configuration file defining the parameters of the different retrieval processes, such as feature extraction, features selection, classification and evaluation.

3.4 Feature Extraction

In this section, popular audio features applied in speech recognition, music information retrieval and environmental sound recognition are discussed. The goal is to identify suitable features for the domain of environmental sounds.

Content-based retrieval usually does not operate on the original data. Instead, features are computed that represent the content more efficiently. For illustration consider one second of an audio file in CD-quality: The original data contain 44100 samples. The first several hundred Mel Frequency Cepstral Coefficients (MFCCs) of the same signal may suffice for retrieval (see Subsection 3.4.8). This is a significant reduction of the amount of data that has to be processed.

There is no widely accepted taxonomy of audio features. A basic approach is to consider the domain of the feature. Time-based features are extracted from the signal in time domain. Spectral features are derived after the signal has been transformed using one of the basic signal processing transforms such as Fourier, Cosine, and Wavelet Transform. Another way to classify audio features is to analyze whether they aim at imitating the human auditory sense. Such features are called *perceptual features*. The author considers features as either time-based or spectral. The ability of a feature to imitate the human auditory sense is regarded as a superordinate property. Time-based features in the investigation comprise of Zero Crossing Rate and Short-Time Energy. Additionally, the author introduces a set of new time-based features that describe the properties of the waveform of the signal. They are Length of High Amplitude Sequence (LoHAS), Length of Low Amplitude Sequence (LoLAS) and Area of High Amplitude (AHA). Spectral features employed are Spectral Flux, Fourier Transform, Cosine Transform, Wavelet Transform, Constant Q Transform, Pitch, Sone, Cepstral Coefficients, Linear Predictive Coding, Perceptual Linear Prediction (PLP) and RASTA-PLP. Perceptive features are Sone, Pitch, PLP, and RASTA-PLP.

3.4.1 Spectral Flux

The Spectral Flux (SF) is the summation of differences between adjacent samples of the signal spectrum in a single frame [20]. It is computed as follows:

$$SF = \sum_{n} \parallel |S[n]| - |S[n+1]| \parallel \tag{4}$$

In the experiments statistical moments of first and second order of the SF for each file are employed.

3.4.2 Fourier Transform

The continuous Fourier Transform (FT) named after Joseph Fourier, is an integral transform that re-expresses a function in terms of sinusoidal basis functions, i.e. as a sum or integral of sinusoidal functions multiplied by some coefficients (*amplitudes*). It offers a frequency domain representation of the signal. The coefficients of the FT may directly be used as a feature. They are also the basis for computations of more complex features (for example MFCC, see Subsection 3.4.8). The FT of a signal is given by Equation 5 and sometimes called the forward FT.

$$F(k) = \int_{-\infty}^{\infty} s(n)\, e^{-2\pi i k n} \mathrm{d}n \tag{5}$$

Equation 6 is called the inverse FT and is used to obtain a reconstruction of the signal in the time domain.

$$s(n) = \int_{-\infty}^{\infty} F(k)\, e^{-2\pi i k n} \mathrm{d}k \tag{6}$$

For digital audio the Discrete Fourier Transform (DFT) is needed. It is defined over discrete, finite or infinite domains. Equations 7 and 8 show the formulae for the calculation of the DFT/Inverse DFT.

$$F(k) = \sum_{n=0}^{l-1} s(n)\, e^{\frac{-2\pi i k n}{l}},\ k = 0, \cdots, l-1 \tag{7}$$

$$s(n) = \frac{1}{l} \sum_{k=0}^{l-1} F(k)\, e^{\frac{2\pi i k n}{l}},\ j = 0, \cdots, l-1 \tag{8}$$

In 1965, Cooley and Tukey [10] first discussed the Fast Fourier Transform (FFT), a DFT algorithm that reduces the complexity of computations for N samples from $O\left(N^2\right)$ to $O\left(N \cdot \log N\right)$. Today, the FFT is a standard technique to compute the FT of a digitized signal.

The first 60 DFT coefficients are used to form a feature vector. Optionally zero-padding is applied to equalize the length of the samples.

3.4.3 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is closely related to the DFT. In contrast to the DFT, which uses complex numbers, the DCT is real-valued. The DCT approximates a signal by a weighted sum of cosine functions with different frequencies. There are several variants of the DCT with slightly modified definitions. The variant DCT-II in Equation 9 is commonly referred to as *the DCT*.

$$f_j = \sum_{n=0}^{N-1} s(n) \cdot \cos\left(\frac{j\pi}{N}\left(n + \frac{1}{2}\right)\right) \tag{9}$$

Equation 10 presents the variant DCT-III which is commonly referred to as *the inverse DCT* (IDCT).

$$s_j = \frac{1}{2} f\left(0\right) + \sum_{k=1}^{N-1} f(k) \cdot \cos\left(\frac{n\pi}{N}\left(j + \frac{1}{2}\right)\right) \tag{10}$$

Similarly to DFT the computation for the DCT is in $O\left(N \cdot \log N\right)$. In practice DCT is often used for lossy data compression (e.g. JPEG) and visual information retrieval. A modified transform, *the modified DCT* is used in MP3 and Vorbis audio compression. This area of application is motivated by the property of the DCT that most of the signal information tends to be concentrated in the low frequency components of the DCT. Because of the lower computational complexity of the DCT, it is employed as an approximation of the Principal Component Analysis (PCA), a linear transform that optimally keeps the subspace that has largest variance, thus decorrelating the input data.

Selected DCT coefficients, the low frequency components, are usable as a feature for classification. Analogously to the DFT the first 60 DCT coefficients are employed for retrieval.

3.4.4 Wavelet Transform

The Wavelet Transform (WT) is a group of time-frequency transforms. It dates back to the early 20^{th} century, when Alfred Haar, a Hungarian mathematician, introduced the first Discrete Wavelet Transform. Generally the WT aims at representing a signal by a finite length or fast decaying oscillating waveform that is scaled and translated to reproduce the signal. This waveform is called the *mother wavelet*. There is a large number of different mother wavelets, the most common ones are *Haar* and *Daubechies* named after Alfred Haar and Ingrid Daubechies [14]. In fact each mother wavelet defines one WT, but for the sake of simplicity one refers to *the WT*. Selection of the optimal mother wavelet depends on the application. Recently, WT started to replace FT in several research and application areas, such as signal processing, speech recognition, and astrophysics.

The drawback ob the FT is, that it does not preserve any spatial information. From the Fourier spectrum we cannot determine where certain frequencies occur in the original signal. The Short-Time Fourier Transform (STFT) aims at tackling this issue. This is performed by dividing the original signal into short frames. Each frame is Fourier transformed. This results in a spectrum that contains local information. The spatial resolution depends on the frame size. With decreasing frame size the spatial resolution increases. At the same time the frequency resolution decreases.

The WT shares these properties of the STFT. Instead of framing the signal, the WT moves a function (the mother wavelet) over the original signal. While the wavelet is translated, it is scaled to match the signal. For each scale and translation value pair, the WT yields a coefficient. The set of all coefficients represent the original signal in terms of the mother wavelet.

Two types of WT exist: Discrete Wavelet Transform (DWT) and continuous Wavelet Transform (CWT). The CWT applies all scales and translations of the mother wavelet. The CWT is given in Equation 11.

$$c(a,b) = \int_{-\infty}^{\infty} s(n)\,\psi(an+b)\,\mathrm{d}n \tag{11}$$

Where a are the scale values and b the translation values. The corresponding coefficient of a and b is $c(a,b)$. CWT is commonly used for signal analysis

in scientific research. It is infinitely redundant but sometimes useful to comprehend particular signal properties. The DWT uses a specific subset of scale and translation values which fulfill the conditions in Equation 12.

$$\psi\left(2^k n + l\right) \text{ with } k, l \in Z \tag{12}$$

DWT is employed in computer science and engineering as a means of signal coding and compression. The DWT is computed by the use of filter banks containing FIR filters. Similar to the FFT, a fast version of the DWT exists, the Fast Wavelet Transform (FWT) [3]. In contrast to the FFT, the computational complexity of the FWT is linear ($O(N)$). Similarly to DCT and FFT coefficients, DWT coefficients are directly employed as features. In the experiments feature vectors containing the first 100 DWT coefficients are used. The mother wavelet employed is the Haar wavelet.

3.4.5 Constant Q Transform

In order to overcome the shortcomings of the Fourier transform for analysis of Western music, Brown introduced the Constant Q Transform (CQT) in [6]. The DFT yields frequency components that are separated by a constant frequency difference and feature constant resolution. These frequency components do not map efficiently to musical frequencies. The Constant Q Transform is similar to the FT but has a constant ratio of center frequency f to resolution δf. Equation 13 illustrates the computation of the CQT:

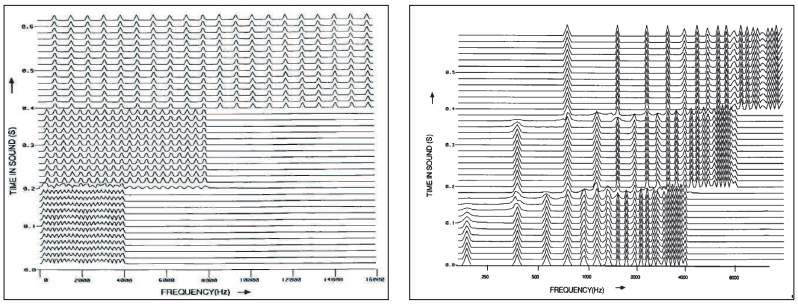
$$X(k) = \frac{1}{M(k)} \sum_{n=0}^{M(k)-1} W(k,n)\, s(n)\, exp\left(\frac{-i2\pi Q n}{M(k)}\right) \tag{13}$$

with:

- window: $W(k,n) = \alpha + (1-\alpha)\cos\left(\frac{2\pi n}{M(k)}\right)$
- variable window width: $M(k) = \frac{SamplingRate \cdot Q}{2^{k/24}}$
- and $Q = \lfloor f/\delta f \rfloor$.

The Constant Q Transform aims to convert the problem of instrument identification or fundamental frequency identification into a straightforward pattern recognition task. The CQT data are transformed against log (frequency). Under this view, sounds with harmonic frequency components

show constant patterns in low frequency space. Figures 5(b) and 5(a) illustrate the presence of this effect with the CQT and its absence with the DFT.



(a) signals transformed with FFT (b) signals transformed with CQT

Figure 5: FT and CQT of three complex sounds having 20 harmonics with equal amplitude [6]. Sounds with harmonic frequency components show constant patterns in low frequency space of the CQT. The FFT lacks this property.

The author utilizes the implementation provided by Brown in [6] using default values to compute CQT coefficients. Mean and variance of the CQT coefficients over each transform window are applied as features.

3.4.6 Pitch

Pitch is the perceptual counterpart of the physical frequency. It is the perceived frequency of a sound. Pitch can not be measured physically, since it is an auditory sensation. Two sounds with measurably different frequencies do not need to have two different pitches, but differences in the perceived pitch implies different frequencies. The author employs a pitch detection algorithm devised by Sun in [56]. For the experiments the maximum bandwidth the algorithm supports is used. Mean and variance of the time dependent pitch are used as features.

3.4.7 Sone

Sone is a unit on a perceptually motivated loudness scale. Loudness is a subjective measure of sound pressure. One phon is defined as the loudness

of a 1 kHz tone at 40 dB SPL (Sound Pressure Level). One sone equals 40 phons. The ratio of sone to phons (1:40) was chosen to represent a doubling of loudness with a doubling in sone. A sound with a loudness of two sone is perceived twice as loud as a sound with loudness one sone. The loudness values of selected frequency bands mapped to sone may be used as features.

For the experiments the MATLAB toolbox of Pampalk is employed [47]. The author computes sone values for 20 frequency bands with a window size of 256 samples. Mean and variance of all sample windows serve as features, hence for each file a 40-dimensional feature vector is obtained.

3.4.8 Cepstral Coefficients

Cepstral Coefficients (CCs) are a popular feature in audio retrieval [37], [65]. The authors of [59] define the cepstrum as the Fourier Transform (FT) of the logarithm (log) of the spectrum of the original signal.

$$signal \rightarrow FT \rightarrow log \rightarrow FT \rightarrow cepstrum$$

In practice, CCs are derived from FFT or DCT coefficients or linear predictive analysis [5]. CCs offer a compact and accurate high order representation of signals. Peaks in the cepstrum correspond to harmonics in the power spectrum.

MFCCs (Mel Frequency Cepstral Coefficients) are an instance of CCs. Computation of MFCCs includes a conversion of the logarithmized Fourier coefficients to Mel scale. After conversion, the obtained vectors have to be decorrelated to remove redundant information. A DCT is applied to receive a decorrelated, more compact representation. In the following sequence the computation of MFCCs is illustrated:

$$signal \rightarrow FT \rightarrow log \rightarrow Mel \rightarrow DCT \rightarrow MFCCs$$

MFCCs are computed using VOICEBOX, a speech processing toolbox for MATLAB [5]. In the experiments the first 20 MFCCs are combined into a feature vector. MFCCs are computed for small signal windows. Hence mean and variance of each coefficient are calculated. Optionally, the author

tries to enhance retrieval quality through the use of delta and double delta features.

BFCCs (Bark Frequency Cepstral Coefficients) are computed similarly to MFCCs. They differ in the applied scale (Bark scale):

$$signal \rightarrow FT \rightarrow log \rightarrow Bark \rightarrow DCT \rightarrow BFCCs$$

Bark scale and Mel scale are perceptually motivated acoustical scales that nonlinearly map the signal frequency. Both nonlinear scales offer higher resolution for low frequencies than for high frequencies.

Again, VOICEBOX is utilized to compute BFCCs. The first 20 BFCCs are selected and their mean and variance is calculated. Additionally, the influence of delta and double delta features is examined.

3.4.9 Linear Predictive Coding

Linear Predictive Coding (LPC) is one of the most powerful speech analysis techniques [49], [58]. The goal of LPC is to estimate the basic parameters of a speech signal, e.g. pitch, formants, spectra, and vocal tract area functions. Formants describe the vocal tract (mouth, throat) of a speaker by its resonances. The formants are extracted by a linear predictor. The linear predictor tries to express the value of a sample by a linear combination of values of previous samples. LPC estimates coefficients using linear prediction, that minimize the mean square error (MSE) between the original signal and the predicted signal. The coefficients of the linear predictor represent the formants of a speech signal. LPC coefficients are employed in speech recognition to distinguish between phonemes. In [45] the authors successfully introduce LPC coefficients to environmental sound recognition in the context of animal sounds. The VOICEBOX implementation is used to obtain LPC coefficients. The first 20 coefficients computed by covariance LPC analysis are employed in the experiments.

3.4.10 Perceptual Linear Prediction

Perceptual linear prediction (PLP) was introduced by Hermansky in 1990 for speaker-independent speech recognition [26]. PLP is based on the concepts

of linear predictive (LP) analysis and additionally emphasizes perceptual issues. LP analysis approximates the original signal in each frequency band equally well. This is not consistent with human hearing where the resolution decreases with increasing frequency. PLP overcomes the shortcomings of LP by implementing several properties of human hearing.

In the first processing step of PLP the windowed audio signal is Fourier transformed. The resulting power spectrum is warped to the Bark scale. The warped spectrum is convolved with an asymmetric critical-band masking curve. The critical-band masking curve approximates the shape of auditory filters. It specifies the spectral resolution of human hearing for each frequency. The resulting spectrum is sampled at approximately 1 Bark intervals. This results in 18 spectral samples for an analysis bandwidth of 0 to 5 kHz (0-16.9 Bark).

The sampled values are weighted by an equal-loudness curve that simulates the sensitivity of human hearing at different frequencies. Cubic-root amplitude compression approximates the power law of hearing that describes the nonlinear relation between the intensity of sound and its perceived loudness in human hearing.

Finally, the spectral samples are approximated by an all-pole model, usually applied in LP analysis. The coefficients of the all-pole model can be used as features directly. Alternatively, they can be further transformed to cepstral coefficients. The computational costs of PLP are similar to those of LP analysis.

The author employs the MATLAB toolbox by Ellis that supports PLP and RASTA-PLP [16]. All 18 coefficients are used in the experiments. PLP coefficients are computed for entire files.

3.4.11 RASTA-PLP

Relative spectral - perceptual linear prediction (RASTA-PLP) is an extension of PLP introduced in [27]. The objective of RASTA-PLP is to make PLP more robust to spectral distortions of the communication channel. RASTA-PLP considers the fact that human perception is sensitive to relative values (changes) and not to the absolute values of a signal. Human

hearing is insensitive to slow variations in the input signal and constant noise introduced by the communication channel. The RASTA technique simulates this by band-pass filtering each frequency channel.

The steps of the RASTA-PLP technique are depicted in Figure 6. From

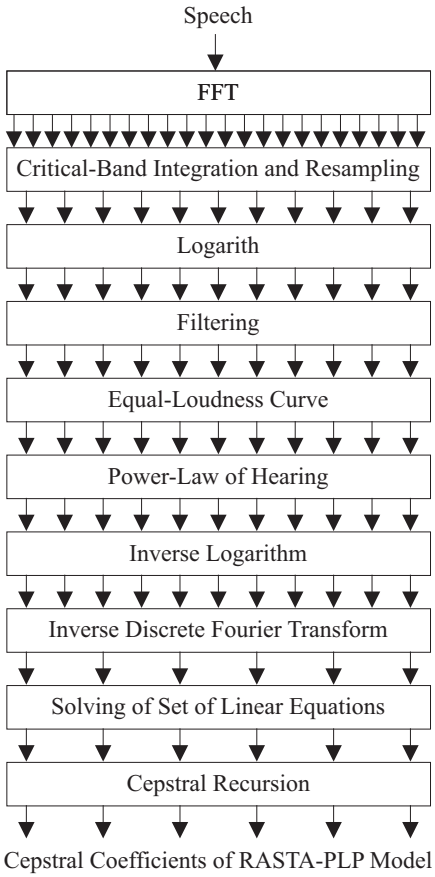


Figure 6: The RASTA-PLP method.

the Fourier Transform of the windowed speech signal, the critical-band spectrum is computed as with PLP. The spectral amplitudes are logarithmized. The log critical-band spectrum is filtered by a band-pass filter. The effect of the band-pass filter is that constant or slowly-varying components in the spectrum are suppressed. Spectral changes below the low cut-off frequency of the filter are ignored in the output. This removes any constant or slowly-varying components from the spectrum. The high cut-off frequency is the upper limit of spectral changes which are preserved. Spectral changes above

the high cut-off frequency of the band-pass filter are suppressed to smooth out artifacts (fast frame-to-frame spectral changes) caused by short-time analysis. Figure 7 illustrates the effect of the band-pass filter on the spectrum. After band-pass filtering the equal loudness curve and cubic-root

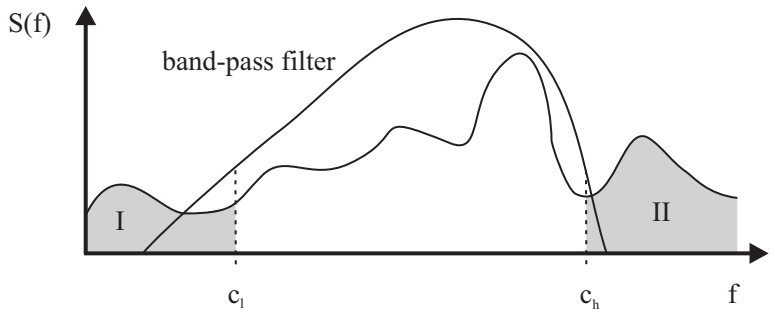


Figure 7: S(f) is the spectrum of a signal s(n). The band-pass filter removes constant and slowly varying components below the low cut-off frequency c_l (area I). Furthermore, artifacts above the high cut-off frequency c_h are removed (area II).

amplitude compression is applied to the relative log spectrum, equivalent to PLP. Prior to the approximation of the spectrum by an all-pole model, the inverse logarithm of the spectrum is computed.

Analogously to the PLP technique, the coefficients or their cepstral coefficients may be employed as audio features. According to Hermansky [27], the RASTA-PLP technique outperforms the PLP technique in applications where the communication channel introduces noise and spectral coloration to the signal (e.g. telephone line). The RASTA technique yields more robust results and decreases error rates in recognition.

In the experiments RASTA-PLP coefficients are computed analogously to PLP coefficients. Again mean and variance of all 76 coefficients, provided by the algorithm of Ellis, are selected for retrieval [16].

3.4.12 Zero Crossing Rate

The Zero Crossing Rate (ZCR) is the number of zero-crossings in the time domain within one second. According to Kedem [31] the ZCR is a measure for the dominant frequency in a signal.

The mean ZCR for entire sample files is used as a feature.

3.4.13 Short-Time Energy

The Short-Time Energy (STE) of an audio signal reflects the amplitude variations over time. The main area of application of STE is the discrimination between silence and non-silence. Equation 14 illustrates the computation.

$$STE = \Delta t \sum_{n=1}^{N} |s\,[n]|^2 \tag{14}$$

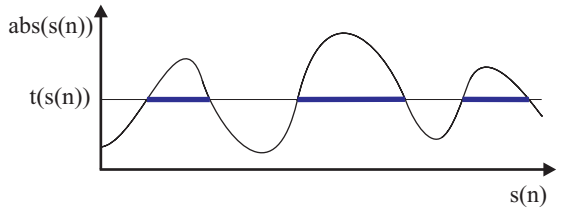
Mean and variance of the STE are computed for entire files.

3.4.14 LoHAS, LoLAS & AHA

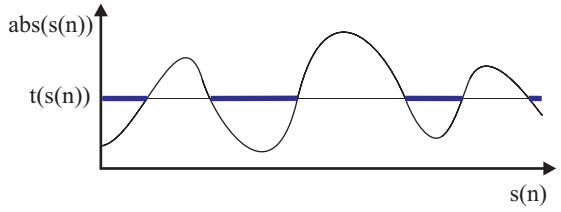
The author introduces a set of new time-based low-level features for audio [45]. The features follow a simple perceptually driven approach. A human observer distinguishes between sounds among other things by the distribution of loud portions and silent portions. Sounds often consist of similar recurrent fragments. Environmental sounds match this concept; footsteps and sirens are repeating sounds. The human auditory sense uses this information to recognize and distinguish between sounds. For example, footsteps differ from a sound of a siren in the repeat rate and the length of the single sounds. On a technical level that means that the high energy segments are different in length. Similarly, the length of pauses between high energy segments contains valuable information.

The introduced features are motivated by this observation. They describe characteristics of the waveform such as peaks and silence. The features are computed based on an adaptive threshold. This threshold separates segments with high amplitudes from segments with low amplitudes in the waveform. The threshold for a particular sound sample is the sum of mean and standard deviation of the absolute sample values. Based on this threshold the length of high amplitude sequences (LoHAS) is computed. The length of a high amplitude sequence represents the number of consecutive samples that have a value greater or equal to the threshold. All LoHAS together represent the distribution of the lengths of high energy segments in the signal. Figure 8(a) illustrates this feature. Analogously, the length of

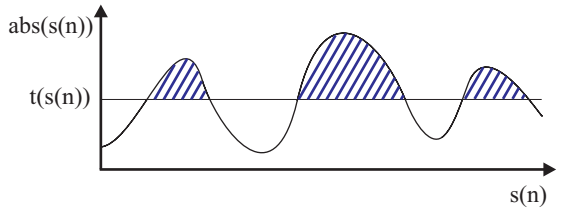
a low amplitude sequence (LoLAS) is defined as the number of consecutive samples that have a lower value than the threshold. The set of LoLAS describes the distribution of lengths of silent segments in the signal. Details are depicted in Figure 8(b). The length of a high amplitude sequence contains temporal information but no information about the loudness of the signal at this section. Sequences with high amplitude can be further characterized by the area below the waveform. The area of high amplitudes (AHA) is the area between the threshold and the signal in a high area sequence. In other words, the AHA feature represents the extent of high energy segments in the signal. Figure 8(c) illustrates this concept.



(a) LoHAS



(b) LoLAS



(c) AHA

Figure 8: LoHAS, LoLAS, and AHA for signal $s\,(n)$ with threshold $t\,(s\,(n))$: (a) Length of High Amplitude Sequence (LoHAS); (b) Length of Low Amplitude Sequence (LoLAS); (c) Area of High Amplitude (AHA).

Statistical properties of LoHAS, LoLAS, and AHA are considered to build features that describe entire sample files. The final features comprise means, standard deviations, and medians of LoHAS and LoLAS over the entire signal. Additionally, the means of AHA are extracted. This results in a feature vector with seven dimensions that is used for classification. LoHAS, LoLAS and AHA are also referred to as Amplitude Descritpor (AD).

3.5 Classification

Classification is an important step in content-based retrieval. The process of classification tries to correctly predict the class of a sample. In this section the classifiers employed in the experiments are described.

There is a large number of classification techniques following different approaches. Statistical methods such as Bayes classification and Gaussian Mixture Models try to estimate the probability density function of the underlying data [28]. Another group of classifiers are learning algorithms that employ artificial intelligence techniques. Some algorithms fit a parametric model to the underlying data. There are supervised learning methods such as Support Vector Machines, neural networks, and non-supervised techniques such as Self-Organizing Maps [34]. A classification technique similar to Self-Organizing Maps is Learning Vector Quantization. Beside parametric techniques (e.g. Support Vector Machines), there are non-parametric techniques such as the Nearest Neighbor.

Three supervised classifiers are selected for the experiments. The simplest way to classify feature vectors is the nearest neighbor rule. The K-NN classifier is employed. The K-NN is a generalization of the nearest neighbor classifier. In the experiments the implementation of Roger Jang is applied [29]. Additionally, the author implemented Learning Vector Quantization (LVQ) using standard MATLAB routines. Finally, a Support Vector Machine (SVM) is applied in classification with different kernels. For this purpose the OSU SVM MATLAB toolbox is used [40].

3.5.1 K-Nearest Neighbor

K-Nearest Neighbor (K-NN) is a popular non-parametric classifier. Details are given in [12]. In contrast to parametric techniques that fit a model to the data or that describe the probability distribution of the data, non-parametric techniques operate on the data directly. The data are a combination of a training set $X = (\mathbf{x}_1, ..., \mathbf{x}_N) \in R^{d \times N}$ containing N training vectors of dimension d and a vector $y = (y_1, ..., y_N) \in R^{1 \times N}$ of corresponding class labels.

The 1-NN (NN) algorithm assigns a new vector $\mathbf{x}$ the class label y_s of the nearest training vector $\mathbf{x}_s$, where

$$s = \arg\min_i \|\mathbf{x} - \mathbf{x}_i\|, \, 1 \leq i \leq N. \tag{15}$$

Similarity in nearest neighbor classification can be measured by any similarity (distance) measure. Frequently, Euclidean distance is used for K-NN. This assignment scheme partitions the feature space according to a Voronoi tessellation. Each cell belongs to one class. Figure 9 illustrates a Voronoi tessellation in two dimensional space. The union of all cells that are assigned to the same class, is the decision region for this class.

The K-NN algorithm with $K > 1$ considers more than just the nearest neighbor for classification. K denotes the number of nearest neighbors of a new feature vector $\mathbf{x}$ that are considered for classification. From these K vectors, k_j vectors belong to class ω_j, with $\sum_{j=1}^{c} k_j = K$ where c is the number of classes. Vector $\mathbf{x}$ is assigned to class ω_i with the greatest number of representatives in the set of K neighbors:

$$i = \arg\max_j k_j, \, 1 \leq i \leq c \tag{16}$$

During training the K-NN classifier learns the training set by rote. Hence, memory and computation costs grow linearly with the size of the training set ($O(N)$).

In the experiments the K-NN classifier is applied with different values for K. The initial value for $K = 1$. K is incremented as long as classification results improve. NN is considered to indicate the quality of the features. Features that discriminate classes well, provide disjoint partitions in the feature space.

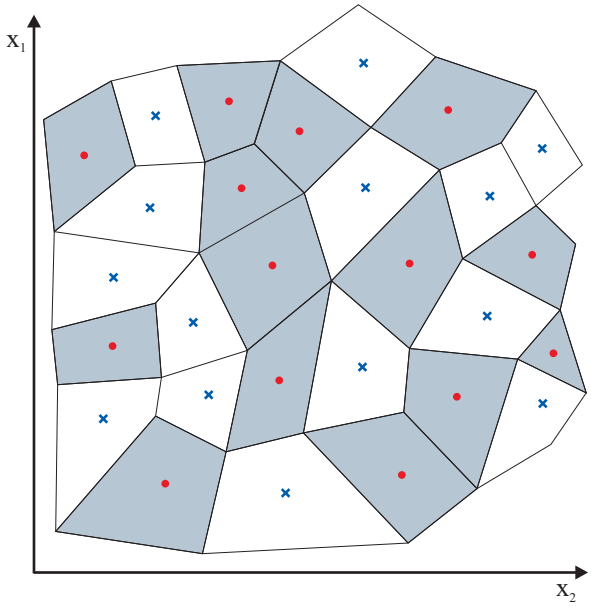


Figure 9: Voronoi tessellation in R^2 of a binary classification problem. Dots are feature vectors of class A, crosses are feature vectors of class B. The gray area is the decision region of class A.

3.5.2 Learning Vector Quantization

Learning Vector Quantization (LVQ) is a classification technique belonging to the basic competitive neural networks. It was introduced by Kohonen [35] and is related to Self-Organizing Maps, also by Kohonen [34].

The LVQ algorithms approximate class distributions of pattern vectors. According to their creator, LVQ algorithms define very good approximations for the optimal decision borders.

Let $\mathbf{x}$ be a sample vector and S_k be the k-th class of an N class classification problem. We first randomly assign a subset of codebook vectors to each class S_k and then search the codebook vector $\mathbf{m}_i$ with the smallest Euclidean distance from $\mathbf{x}$. It is possible to perform this assignment without intermixing codebook vectors that belong to different classes, even if the class distributions overlap. The sample $\mathbf{x}$ is thought to appertain to the same class as the closest $\mathbf{m}_i$. The decision border is defined by the codebook vectors closest to the class border. The $\mathbf{m}_i$ have to be placed into the signal

space in such a way that the nearest-neighbor rule minimizes the average expected misclassification probability.

Let

$$c = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \tag{17}$$

define the index of the nearest $\mathbf{m}_i$ to $\mathbf{x}$. Let $\mathbf{x} = \mathbf{x}(t)$ be a time-series sample of input, and let the $\mathbf{m}_i(t)$ represent sequential values of the $\mathbf{m}_i$ in the discrete-time domain. LVQ1, the basic Learning Vector Quantization process is given in Equations 18 to 20:

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \qquad \mathbf{x}, \mathbf{m}_c \in S_k \tag{18}$$

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \quad \mathbf{x} \in S_i, \mathbf{m}_c \in S_j,\ i \neq j \tag{19}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t), \qquad i \neq c \tag{20}$$

The asymptotic values of $\mathbf{m}_i$ obtained in the above process define a vector quantization for which the rate of misclassification is approximately minimized. The learning rate $\alpha(t)$ is usually made to decrease monotonically with time. Kohonen recommends an $\alpha < 0.1$. The exact law $\alpha = \alpha(t)$ is not crucial. If only a restricted set of training samples is available, they may be applied cyclically, and $\alpha(t)$ may even be made to decrease linearly to zero.

The basic LVQ algorithm is illustrated in Figure 10.

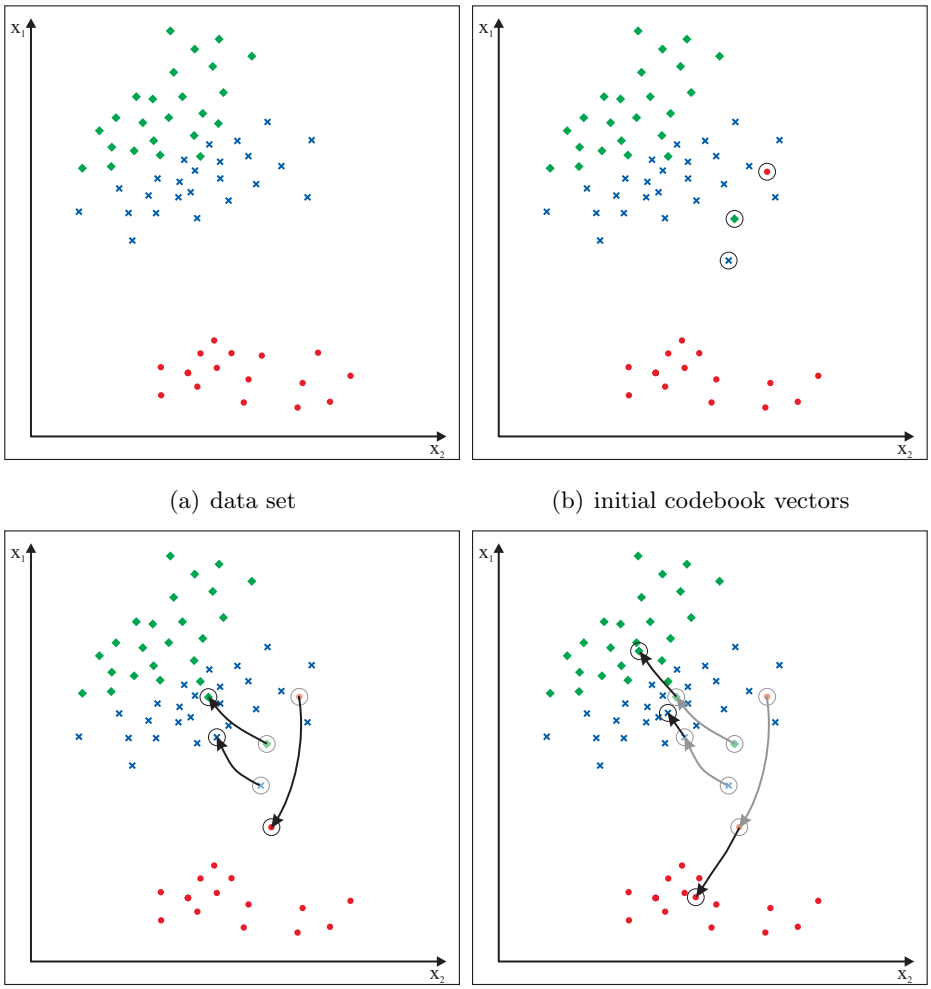
The optimized-learning-rate LVQ1 (OLVQ1) is an improved version of the LVQ1 presented above. OLVQ1 differs from LVQ1 in a way that it uses an individual learning rate $\alpha_i(t)$ that is assigned to each $\mathbf{m}_i$. Let c be defined in Equation 17, and let $f(\mathbf{x}) = +1$, $f(\mathbf{x}) = -1$ denote correct respectively incorrect classification of $\mathbf{x}$. Equations 21 to 23 define the new learning process:

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha_c(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \quad f(\mathbf{x}) = +1 \tag{21}$$

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha_c(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \quad f(\mathbf{x}) = -1 \tag{22}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t), \qquad i \neq c \tag{23}$$

If all samples are used with equal weight, the statistical accuracy of the learned codebook vectors is approximately optimal. OLVQ1 is not the only derivative of LVQ algorithm, several others exist (LVQ2, LVQ3, etc.) [33].



(a) data set (b) initial codebook vectors

(c) after 25 learning steps (d) after 50 learning steps

Figure 10: The learning process of the LVQ classifier: (a) the original data set with shape-coded class labels; (b) the circles mark the initial codebook vectors for each class; (c) and (d) display the path of the codebook vectors while they move towards the group of training patterns that belong to the same class.

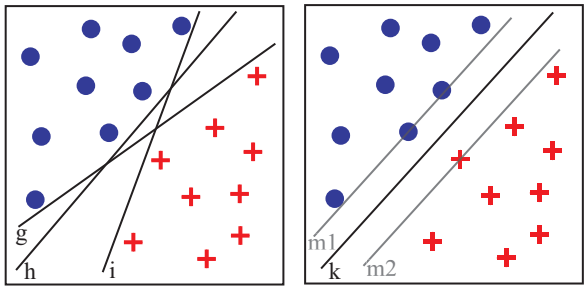
Kohonen suggests the use of the same number of codebook vectors for each class. The upper limit of the total number of codebook vectors is determined by time and computational constraints.

In the experiments, an LVQ with eight hidden neurons, a learning rate of 0.01 and 200 epochs is used. The classifier is supplied with the distribution of classes in the training set.

3.5.3 Support Vector Machines

Support Vector Machines (SVMs) are supervised, statistical learning methods applicable for classification and regression [4], [61]. They are also known as maximum-margin classifiers.

Given two separable clouds of points $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_k, y_k)$ where $\mathbf{x}_i \in R^n$ and $y_i \in \{-1, +1\}$, an SVM constructs an optimal separating hyperplane $\mathbf{wx} + b = 0$, that maximizes the distance between the hyperplane and the nearest data points of each cloud (these points are the support vectors). The distance between the support vectors and the hyperplane is called margin. Figure 11 depicts the difference between a suboptimal and an optimal separating hyperplane.



(a) (b)

Figure 11: Optimal Separating Hyperplanes (OSH): (a) g, h, i are valid but not optimal separating hyperplanes. (b) k is the OSH, the distance between k and m1 respectively m2 is equal and maximal.

The hyperplane is not constructed in feature space. Instead the saddle point of the following Lagrange functional is calculated:

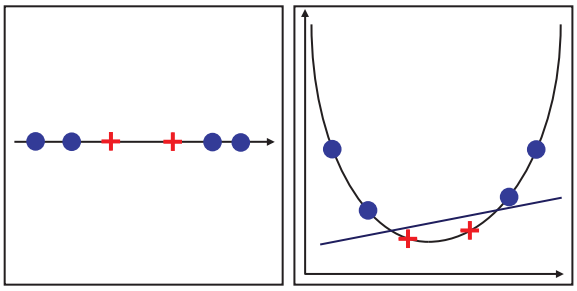
$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \{y_i [(\mathbf{w} \cdot \mathbf{x}) + b] - 1\}, \qquad (24)$$

where α_i are the Lagrange multipliers. Equation 24 may be transformed into problem 25 which is easier to solve,

$$\bar{\mathbf{w}} = \sum_{i=1}^{l} \bar{\alpha_i} y_i \mathbf{x}_i, \qquad \bar{b} = -\frac{1}{2}\bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s] \tag{25}$$

where $\mathbf{x}_r$ and $\mathbf{x}_s$ are two arbitrary support vectors with $\bar{\alpha}_r, \bar{\alpha}_s > 0, y_r = 1, y_s = -1$. Slack variables ζ_i and a penalty function $F(\zeta) = \sum_{i=1}^{l} \zeta_i$ are the means by which SVMs become applicable for the non-separable case [11]. The separating hyperplane is constructed in such a manner that the number of falsely classified $\mathbf{x}_i$ is minimal. This consequently minimizes $F(\zeta)$. The slack variables only influence the Lagrange multipliers α_i. Hence, the solution for the optimization problem stays the same as for the separable case.

In practice, most problems are not linearly separable. Instead of identifying a non linear separating function, the data points are transformed into a higher order space in which they become linearly separable. This is achieved by the use of kernels. Figure 12 illustrates the effect of a polynomial kernel that maps the input space into a feature space of higher order.



(a) not linearly separable (b) linearly separable

Figure 12: The kernel maps the one-dimensional input space (a) into a feature space of higher dimensionality, where the inputs become linearly separable (b).

Equation 26 describes the SVM classifier, where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel used.

$$f(x) = sign\left(\sum_{support\ vectors} \bar{\alpha_i} y_i K(\mathbf{x}_i, \mathbf{x}_j) + \bar{b}\right) \tag{26}$$

There are three typical kernel functions:

1. polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = [(\mathbf{x}_i \cdot \mathbf{x}_j) + 1]^d$,
2. Radial Basis Function (RBF):
   $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^2 / 2\gamma^2\right)$, and
3. sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(scl \cdot (\mathbf{x}_i \cdot \mathbf{x}_j) - off)$,

where *scl* (scale) and *off* (offset) are parameters that have to be chosen with care. The kernel becomes invalid for particular parameter values.

Kernel functions are not limited to the ones mentioned above. Any continuous symmetric non-negative definite function (Mercer's Theorem) is a valid kernel function [2].

Beside K-NN and LVQ, an SVM classifier is applied in the experiments. Since there is no method to determine the optimal kernel function, different kernels are tested. Additionally to a linear kernel, polynomial kernels of second and third order and an RBF kernel are employed.

4 Results

In this section the results of the experiments are presented. For easier understanding we distinguish between four levels of retrieval quality:

$$very\,good > good > mediocre > poor.$$

Table 1 lists the four levels and their numerical correspondents. As discussed

very good	≥ 80 %
good	< 80 %
mediocre	< 60 %
poor	< 40 %

Table 1: The four levels of retrieval quality and their numerical correspondents.

in Subsection 2.3, computation of Recall and Precision relies on the set of retrieved documents. In this thesis the set of retrieved documents is always the entire test set. High Recall values indicate high recognition rates for the classes. High Precision values suggest that only a small number of documents not belonging to the class are retrieved (false positives). A class with high Precision values is well separated from the others. Retrieval quality is regarded to be very good if Precision *and* Recall are $\geq 80\%$. If Recall is 91% and Precision is 30% the overall quality is considered to be *poor*. Note that this quantization is valid only for this specific dataset.

The following parameters were used for the classifiers: K-NN classification was performed with $k \in \{1, 2, 3\}$. LVQ used a learning rate $\alpha = 0.01$ and 200 epochs for training. SVM was tested with polynomial kernels of first, second and third order as well as an RBF kernel with $\gamma = 0$.

Subsections 4.1.1 to 4.1.7 address retrieval results obtained with the previously discussed features and classification algorithms. In Subsection 4.3 the results for one optimized combination of features and the NN classifier are shown. Finally, in Subsection 4.5 performance of the three classifiers is discussed.

4.1 Individual Features

As mentioned above the experiments were first conducted with all features separately. Starting with the basic signal processing transforms in Subsection 4.1.1 and ending with the cepstral coefficients in Subsection 4.1.7. This section will list only the optimal results that were achieved using the above mentioned classifiers. The features are loosely grouped by performance and type. It is noteworthy that several features designed for speech processing perform fairly well in the domain of environmental sounds.

4.1.1 Basic Signal Processing Transforms

The basic signal processing transforms, namely DWT, CWT, FFT, and DCT, yield mediocre results at best. Coefficients of the Discrete Wavelet Transform (using a Haar motherwavelet and 100 coefficients) are no discriminative feature for the used dataset (see Table 2).

DWT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	100%	19%	100%	19%	0%	0%
crowds	0%	0%	0%	0%	0%	0%
footsteps	0%	0%	0%	0%	0%	0%
signal	0%	0%	0%	0%	100%	17%
thunder	0%	0%	0%	0%	0%	0%

Table 2: Results (recall and precision) of DWT for each class (rows) obtained by different classifiers (columns).

All three classifiers fail. They are not able to discriminate the classes. All samples are assigned to one single class. Coefficients of the continuous Wavelet transform perform slightly better, though results stay disappointing (see Table 3).

The best performing classifier is the NN algorithm. Results are poor for all classes except for crowds where a mediocre level is reached. LVQ and SVM fail to distinguish between the classes; all samples are assigned the same class label.

CWT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	48%	16%	100%	19%	0%	0%
crowds	43%	46%	0%	0%	0%	0%
footsteps	14%	41%	0%	0%	0%	0%
signal	10%	18%	0%	0%	100%	17%
thunder	0%	0%	0%	0%	0%	0%

Table 3: Results (Recall and Precision) of CWT for each class (rows) obtained by different classifiers (columns).

The most suitable classifier with the DCT coefficients feature is the LVQ technique. Performance of the LVQ is slightly above the performance of the NN method. The SVM classifier achieves only poor results. It does not distinguish between three of the five classes. Cars, crowds and signals are regarded as one class as illustrated in Table 4.

DCT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	43%	44%	76%	51%	0%	0%
crowds	68%	48%	84%	50%	30%	64%
footsteps	25%	46%	20%	62%	0%	0%
signal	60%	40%	38%	89%	95%	20%
thunder	28%	47%	49%	54%	36%	67%

Table 4: Results (Recall and Precision) of DCT for each class (rows) obtained by different classifiers (columns).

FFT coefficients used as features perform comparable to other coefficients of basic signal processing transforms. Table 5 illustrates the results that indicate that FFT coefficients may be used in combination with other (better performing) features, but are not discriminative to be used as a reliable feature for classification.

The results described above are not surprising. With the basic signal processing transforms, information in the high frequency bands is lost.

FFT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	74%	50%	58%	53%	32%	38%
crowds	52%	44%	50%	49%	26%	42%
footsteps	8%	40%	8%	69%	32%	40%
signal	60%	52%	59%	59%	56%	43%
thunder	33%	31%	56%	29%	61%	37%

Table 5: Results (Recall and Precision) of FFT for each class (rows) obtained by different classifiers (columns).

Therefore discrimination is based on the low frequency bands, that obviously do not carry enough discriminative information.

4.1.2 Constant Q Transform

The CQT originally was introduced as a replacement for the FT that is better suited for analysis of western music. With the SVM and LVQ algorithms, the results are disappointing. In contrast to that, the NN technique performs well. Uniformity inside the classes crowds and thunder results in Precision values above 70%. However results are far from satisfactory for practical use. As can be seen in Table 6 three classes are classified well, i.e. recall $> 60\%$ in combination with precision $> 60\%$. The two remaining classes show mediocre and poor performance.

CQT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	60%	60%	100%	19%	0%	0%
crowds	91%	72%	0%	0%	0%	0%
footsteps	89%	61%	0%	0%	0%	0%
signal	38%	100%	0%	0%	100%	17%
thunder	51%	78%	8%	100%	0%	0%

Table 6: Results (Recall and Precision) of CQT for each class (rows) obtained by different classifiers (columns).

4.1.3 ZCR, STE, and SF

The Zero Crossing Rate is an indicator for the fundamental frequency. Table 7 shows the results for the ZCR feature. Results are poor throughout the classifiers, this supports the assumptions that the critical information for distinguishing between the considered five classes of environmental sounds is in the high frequency bands.

ZCR	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	40%	22%	0%	0%	3%	60%
crowds	33%	37%	0%	0%	1%	25%
footsteps	15%	21%	0%	0%	1%	100%
signal	32%	20%	0%	0%	98%	17%
thunder	1%	14%	100%	19%	0%	67%

Table 7: Results (Recall and Precision) of ZCR for each class (rows) obtained by different classifiers (columns).

STE performs poorly. Since Short-Time Energy is a short time feature one may expect poor performance in an environment where analysis is performed on entire files. STE is applicable in frame-based approaches where sound files are considered as series of short (overlapping) sound frames. Surprisingly, the NN classifier yields mediocre results for two classes, namely footsteps and signal (see Table 8).

STE	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	30%	18%	0%	0%	0%	0%
crowds	19%	37%	0%	0%	0%	0%
footsteps	70%	54%	0%	0%	0%	0%
signal	58%	50%	100%	17%	100%	17%
thunder	9%	24%	0%	0%	0%	0%

Table 8: Results (Recall and Precision) of STE for each class (rows) obtained by different classifiers (columns).

The Spectral Flux has hardly any discriminative power. Except for crowds which are classified comparably to the NN and the SVM techniques (see Table 9). The LVQ algorithm fails. Again all files are regarded to belong to one single class. One may argue that this is due to the fact that analyis is performed for entire sound files. The spectral fluctuation inside a sound file may only contain useful information if the sound has tone-like structure. In order to achieve this, frame-based analysis has to be performed. Sufficiently short frames, should contain sound information that could be regarded as one tone.

SF	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	16%	23%	0%	0%	0%	0%
crowds	67%	59%	0%	0%	75%	59%
footsteps	16%	38%	0%	0%	1%	17%
signal	73%	31%	100%	17%	84%	21%
thunder	32%	53%	0%	0%	9%	53%

Table 9: Results (Recall and Precision) of Spectral Flux for each class (rows) obtained by different classifiers (columns).

4.1.4 Pitch, PLP, and RASTA-PLP

Pitch is the perceptual counterpart of the physical dominant frequency. Results are poor throughout the classifiers. The NN technique performs best, managing to discriminate one class with an acceptable precision, namely signals (see Table 10).

PLP as well as RASTA-PLP show mediocre performance. Results are not consistent; some classes are separated well while others are not. This is true for NN and SVM classifiers. LVQ fails with both features as illustrated in Table 11 and Table 12. Features that are based on linear predictive coding (such as PLP and RASTA-PLP) perform better, the less they are optimized for speech. This becomes evident when results of the LPC coefficients (see Table 14) are compared with the results in Tables 11 and 12.

Pitch	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	46%	46%	0%	0%	23%	91%
crowds	63%	51%	100%	24%	26%	50%
footsteps	33%	42%	0%	0%	24%	86%
signal	78%	57%	0%	0%	88%	46%
thunder	29%	46%	0%	0%	92%	38%

Table 10: Results (Recall and Precision) of Pitch for each class (rows) obtained by different classifiers (columns).

RASTA-PLP	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	55%	41%	0%	0%	19%	41%
crowds	54%	51%	100%	24%	88%	41%
footsteps	45%	62%	0%	0%	42%	52%
signal	32%	54%	0%	0%	31%	66%
thunder	41%	33%	0%	0%	27%	32%

Table 11: Results (Recall and Precision) of RASTA-PLP for each class (rows) obtained by different classifiers (columns).

It is noteworthy, that PLP coefficients yield slightly better classification results than RASTA-PLP coefficients. This indicates that the optimizations for speech integrated in RASTA-PLP decrease performance for environmental sounds.

4.1.5 LoHAS, LoLAS & AHA

It was not possible to achieve consistently good retrieval results with the amplitude-describing features introduced in [45]. Results are disappointing as can be seen in Table 13. LoHAS, LoLAS and AHA (short Amplitude Descriptor AD) do not contain enough discriminative information that would make them useful for classification as the single discriminative feature. Analysis of the waveforms of the data used explains the achieved performance. With the selected classes of environmental sounds, discriminative

PLP	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	59%	50%	0%	0%	77%	50%
crowds	56%	53%	100%	24%	63%	58%
footsteps	33%	52%	0%	0%	37%	48%
signal	68%	73%	16%	100%	37%	79%
thunder	70%	56%	0%	0%	64%	60%

Table 12: Results (Recall and Precision) of PLP for each class (rows) obtained by different classifiers (columns).

information is encoded in the amplitude. LoHAS, LoLAS and AHA use an adaptive threshold that destroys large portions of this important amplitude information.

AD	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	53%	34%	42%	61%	39%	47%
crowds	57%	50%	52%	50%	61%	53%
footsteps	60%	74%	75%	68%	55%	83%
signal	49%	52%	65%	41%	81%	52%
thunder	29%	52%	36%	57%	59%	66%

Table 13: Results (Recall and Precision) of the amplitude describing features LoHAS, LoLAS & AHA for each class (rows) obtained by different classifiers (columns).

4.1.6 LPC and Sone

LPC coefficients represent the formants of a speech signal. Use of LPC coefficients in the context of environmental sound recognition is debatable. Nevertheless, the author incorporated LPC in the experiments. The obtained results are presented in Table 14. The NN classifier yields inconsistent results for the LPC coefficients. The quality indicators range from poor to very good. Results obtained from the LVQ classifier are comparable

to those of the NN but slightly more consistent. The SVM algorithm performs equally well to the other classifiers, with a linear kernel. Other kernels yield worse results. Surprisingly, classification with LPC coefficients yields results on a higher quality level than classification with other features that are theoretically more suitable for environmental sound recognition.

LPC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	68%	83%	73%	67%	76%	72%
crowds	82%	73%	72%	69%	55%	82%
footsteps	39%	63%	42%	67%	57%	66%
signal	67%	92%	68%	65%	79%	60%
thunder	84%	49%	78%	61%	79%	63%

Table 14: Results (Recall and Precision) of LPC for each class (rows) obtained by different classifiers (columns).

The Sone feature contains perceived loudness information of 40 frequency bands. The results obtained by this feature are shown in Table 15. Performance is sufficient for this dataset. With the NN classifier results around 70% recall are obtained. This is a satisfactory performance for environmental sounds. The LVQ algorithm fails due to the assignment of all samples to a single class. The SVM classifier yields very good results for crowds and footsteps, good results for cars and thunder and mediocre results for signals.

Sone	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	70%	66%	0%	0%	78%	78%
crowds	70%	80%	0%	0%	83%	74%
footsteps	79%	94%	0%	0%	91%	84%
signal	75%	72%	100%	17%	43%	70%
thunder	76%	61%	0%	0%	78%	71%

Table 15: Results (Recall and Precision) of Sone for each class (rows) obtained by different classifiers (columns).

4.1.7 BFCC and MFCC

The cepstral coefficients, namely BFCCs and MFCCs, yield predominantly good and very good results shown in Table 16 and Table 17. Results are consistent throughout all classifiers. All classification methods generate high recall precision pairs.

BFCC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	52%	80%	67%	72%	65%	67%
crowds	96%	87%	80%	84%	94%	82%
footsteps	93%	90%	80%	87%	92%	86%
signal	68%	96%	65%	80%	59%	89%
thunder	92%	63%	78%	56%	71%	66%

Table 16: Results (Recall and Precision) of BFCC for each class (rows) obtained by different classifiers (columns).

MFCC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
cars	56%	79%	47%	56%	59%	66%
crowds	92%	87%	95%	80%	83%	93%
footsteps	92%	87%	69%	78%	82%	93%
signal	65%	96%	54%	72%	79%	89%
thunder	87%	60%	72%	57%	83%	56%

Table 17: Results (Recall and Precision) of MFCC for each class (rows) obtained by different classifiers (columns).

It is noteworthy, that the very similar results are a consequence of the computation methods of BFCCs and MFCCs. Subsection 3.4.8 gives the explanation: BFCCs and MFCCs differ only in the perceptual scale they use. The information they contain is virtually the same. As the best performing features, BFCCs and MFCCs are candidates for the basis of feature combinations discussed in the next section.

4.2 Preliminary Summary

In the previous sections, multiple features were evaluated individually for the purpose of environmental sound recognition. The results obtained are satisfactory. While simple features, such as coefficients of basic time to frequency transforms (DFT, DWT, and DCT), are not able to capture discriminative properties of the classes, features of high complexity, such as cepstral coefficients, are well suited for recognition of environmental sounds. Beside the complexity, quality of retrieval often depends on the dimensionality of the features. While low-dimensional features, such as ZCR are limited in their expressiveness, high-dimensional features provide more explanatory power. For example, retrieval quality of MFCCs improves with increasing number of selected components.

The classifiers, selected for the experiments, perform differently for the individual features. Although SVM is a sophisticated technique compared to K-NN, results of K-NN are comparable to those of the SVM. This results from the low dimensionality of the feature vectors. For high-dimensional feature vectors the performance of K-NN decreases, while the SVM often benefits from additional dimensions. The results of LVQ are not as consistent as the results of the other classifiers. LVQ is not able to explain poor features. For example, the feature data of DFT or Pitch are too complex for LVQ. The reason for this may be the simple classification scheme of LVQ applied in the experiments. Since every class is represented by only one codebook vector, the available information about a class reduces to a single data point in feature space. During training, the test samples are assigned to the class of the nearest codebook vector. This classification scheme is too simple to explain the complex structure of several features in the experiments. Good results of LVQ indicate that each class forms a single cluster in feature space. Due to this property, LVQ is well suited for analyzing the structure of high-dimensional data.

There are different reasons for the poor retrieval quality of some features in the experiments. The coefficients of DCT, DFT, and DWT, contain only information of low frequency bands. High frequencies, that are necessary to characterize certain environmental sounds, are neglected. This explains the

poor quality of retrieval, obtained by the transform coefficients. Pitch and ZCR suboptimally perform in the experiments. Due to their low-dimension, they fail to explain test and training data. Nevertheless, these features may improve retrieval quality in combination with other features (see Section 4.3).

The experiments presented comprise of features traditionally applied in speech recognition, such as LPC, PLP, and RASTA-PLP. While PLP and RASTA-PLP yield moderate retrieval results, LPC coefficients outperform most of the other features employed. All classifiers yield consistent results for the LPC feature. This indicates that the LPC coefficients are highly discriminative for environmental sounds in the experiments. Even classification by LVQ yields high Recall and Precision values, which confirms the assumption that LPC coefficients cluster the feature space according to the classes of the data set.

Cepstral coefficients (MFCCs and BFCCs) perform comparably to LPC. Most information is contained in the first few coefficients. That allows for low-dimensional but expressive feature vectors. Both K-NN and LVQ perform well for cepstral coefficients. Consequently, the structure of cepstral coefficients is easy to explain.

The Amplitude Descriptor (AD) yields inconsistent results. In contrast to cepstral coefficients and LPC, the AD comprises only time-based features. In general, a combination of spectral and time-based features is promising because it explains different aspects of the signal. While spectral features characterize frequency characteristics, time-based features incorporate temporal information and loudness. In Section 4.3, the author combines features of different domains in order to improve the recognition rate.

4.3 Combined Features

In this section results of feature combinations are discussed. Table 18 shows the results for one such combination of features.

Empirically, features were combined and removed from the combination until an optimal solution was found for the NN classifier with $K = 1$. The basis for the feature combination (FC) are the first 13 MFCCs. All 20 LPC

Com-	K-NN		LVQ		SVM	
bination	Recall	Precision	Recall	Precision	Recall	Precision
cars	82%	85%	75%	67%	76%	86%
crowds	99%	88%	3%	75%	97%	89%
footsteps	90%	97%	78%	86%	94%	93%
signal	79%	94%	90%	33%	65%	95%
thunder	87%	77%	60%	92%	88%	70%

Table 18: Results (Recall and Precision) of one optimized feature combination composed of for each class (rows) obtained by different classifiers (columns).

coefficients, five statistical moments of first and second order of the LoHAS, LoLAS & AHA features, the mean Spectral Flux, the first RASTA-PLP coefficient and the mean sone value comprise the entire 41-dimensional FC vector.

4.4 Data Analysis

In order to gain further insights into the structure of the feature data Principal Component Analysis is performed. The correlation matrix is used for computation and Varimax rotation is applied. The Varimax rotation rotates a factor loading matrix in a way that preserves the orthogonal positions of the eigenvectors. In PCA, Varimax Rotation is used to maximize the sum of the variances of the factor loadings [55], [30]. Factor loadings are in the interval between -1 and 1. The factor loadings represent the amount of information of a Principal Component that is contained in a feature component. High factor loadings (high absolute values) indicate that significant information of the corresponding Principal Component is contained in the feature component. A load of zero indicates that the feature component contains no information represented by the corresponding factor.

The Principal Component Analysis of the feature set identifies principal components describing 72% of the data (21 components explain 90% of the data). The first Principal Component explains around 12% of the variance. These facts support the hypothesis that the components of the

feature combination (FC) do not depend heavily on one another. Table 19 lists a selected portion of the Factor Loading Matrix. Only the Principal Components that are loaded significantly ($|loading| \geq 0.70$) by two or more feature componentes are listed.

	Principal Component				
Feature	1	2	3	5	7
mean LoHAS	**0.93**	-0.20	0.01	-0.03	-0.05
mean LoLAS	0.59	-0.19	0.37	0.13	-0.02
AHA mean	**0.86**	0.09	0.05	0.04	-0.17
median LoHAS	**0.93**	-0.04	-0.07	-0.01	0.11
median LoLAS	0.53	-0.19	-0.26	-0.19	0.13
SF	0.00	0.68	-0.06	-0.33	-0.03
LPC 1	**-0.73**	0.13	0.30	0.28	-0.16
LPC 6	-0.15	**0.77**	0.00	0.16	0.00
LPC 7	0.03	0.00	0.03	-0.08	**0.73**
MFCC 3	0.23	-0.42	0.00	-0.05	**-0.70**
MFCC 4	0.26	**-0.73**	0.18	-0.14	-0.26
MFCC 5	-0.01	-0.28	0.42	-0.13	-0.62
MFCC 6	0.08	-0.36	**0.76**	-0.19	-0.02
MFCC 7	-0.10	0.17	**0.76**	-0.09	-0.01
MFCC 10	-0.08	-0.02	0.02	**0.81**	0.14
MFCC 11	-0.08	0.18	-0.03	**0.77**	0.09
RASTA-PLP	-0.01	0.06	-0.07	0.57	-0.23
Sone	0.40	-0.16	-0.52	-0.40	-0.35

Table 19: Selected parts of the Factor Loading Matrix for the feature combination.

Three components of the AD highly load on the first Principal Component indicating a certain level of redundancy. It is not surprising that the median and the mean of LoHAS/LoLAS are correlated. Two pairs of MFCC significantly load two Principal Components, but factor loadings are not high enough to justify speaking of redundancy. Principal Components two and seven are loaded by one LPC coefficient and one MFCC, respec-

tively. The MFCC and LPC coefficients are correlated, they load the same components. Generally, LPC coefficients show low redundancy. Eight LPC coefficients exhibit significant factor loadings, all eight load different Principal Components.

Data analysis proves that the selected feature combination is suited well for the given data, redundancy is low. For more general statements about the quality of the feature combination more test data are needed.

4.5 Comparison of Classifiers

As expected, classification quality does not depend primarily on the classifier but on the feature. In general, the three classifiers K-NN, LVQ, and SVM, perform comparably with good features, though there is a disadvantage for the LVQ algorithm. In several cases the LVQ technique fails to explain the underlying distribution of the data (for illustration see Subsection 4.1.4 and Subsection 4.1.6). This is due to the employed implementation using one codebook vector per class. Thus, the complexity of the decision boundary is limited.

SVM differs from the other classifiers applied in the experiments. K-NN and LVQ depend on the clustering of samples in feature space. They deliver satisfactory results when the classes form disjoint clusters. In contrast, SVM constructs a more abstract parametric model, such as a linear or polynomial model, depending on the kernel used. As a consequence SVM depends less on the distribution of samples in feature space. A model of low order tends towards better generalization ability, while with a model of high order, classification runs the risk of overfitting.

SVM and K-NN perform comparably in the experiments. There is no clear winner. For high dimensional feature vectors, SVM usually outperforms K-NN. SVM is a sophisticated classification technique that exhibits its strengths with high-dimensional data (hundreds of dimensions). Performance of K-NN decreases with high-dimensional data. In the experiments relatively low-dimensional feature vectors are employed. This may be the reason for the similar performance of SVM and K-NN in the presented investigations.

In most experiments K-NN is used with $K = 1$. The K-NN with different $K > 1$ is tested to identify an optimal classification. In the majority of cases K of one yields the best results.

The author employs different SVM kernels for the discrimination of environmental sounds. The linear kernel is well suited for most features. That indicates a clear structuring of the feature data. Polynomial kernels are outperformed by linear and RBF kernels.

Computational complexities of the investigated classifiers are different. SVM and K-NN can be trained faster than the LVQ algorithm (less than 0.2 seconds versus 15 seconds). There is no significant difference in the time used for classification. Classification is performed in less than 0.2 seconds. All three classifiers are well suited for frame-based classification. Furthermore, they may be employed in mobile respectively real time applications.

5 Related Work

There are few studies that survey techniques for environmental sound recognition. An unbiased survey should consider methods from various research fields dealing with audio. This is what this thesis tries to achieve.

There are different types of audio retrieval techniques. Numerical representation of signals by features is common to all methods. Approaches can be grouped by the way similarity among different signals is detected. A straight-forward technique is to apply a distance measure directly to the features. Pioneering work in this area concerning audio is performed in [64]. The authors develop a content-based audio retrieval system (Muscle Fish) that distinguishes classes such as animals, machines, musical instruments, telephone, etc. They extract features such as loudness, pitch, brightness and bandwidth. Similarity is measured using a weighted Euclidean distance (Mahalanobis distance). Classification is accomplished by the nearest neighbor rule. An alternative to directly measuring similarity is the use of artificial intelligence techniques such as Support Vector Machines (SVM) [11], Hidden Models (HMM) or Artificial Neural Networks (ANN). An early example in the domain of audio processing is presented in [18]. The authors apply a self-organizing neural network to cluster similar sounds. Another way of classification is based on template matching [21]. The author extracts MFCC features from the audio signal and clusters the feature space into distinct cells with a quantization tree (Q tree). Histograms are considered as templates. They represent the distribution of feature vectors over the partitions of the tree. Templates are compared by distance measures (e.g. Euclidean distance or cosine distance).

Segmentation is an important preprocessing step of audio analysis. It is employed to discriminate different types of sound such as speech, music, environmental sounds and combinations of these. The authors of [53] separate music and speech with low level features. They apply Spectral Centroid, Spectral Flux (SF), Zero Crossing Rate (ZCR), Spectral Roll-off, and Percentage of Low Energy Frames to represent the audio signal. Different classification techniques such as Gaussian Mixture Model (GMM) and Nearest Neighbor (NN) are used to separate speech from music based on

these features [28]. The same task is accomplished in [7] using a different set of features (e.g. Amplitude, Cepstra, and Pitch).

A more comprehensive study on audio segmentation is necessary to separate environmental sounds from speech and music. In [68] the authors successfully separate speech, music, song, environmental sounds and some selected combinations of these sound types. Features for this purpose include Energy, ZCR, Fundamental Frequency, and Spectral Peak.

Based on successful segmentation of an audio stream, different audio types can be further analyzed. The most intensive research took place in the area of speech recognition. Beside classical recognition of speech [48], researchers focus on recognition of the spoken language [46]. Another field of research is classification of the speaker (e.g. for customization issues or authentication) [50]. In the area of multimodal dialog systems, recognition of human emotions from audio gains focus [8].

Not only speech recognition but music information retrieval (MIR) also gained importance through the availability of huge amounts of digital music. MIR consists of classification and structural analysis. Classification concerns recognition of instruments, artists and genres. A number of speech recognition features are applicable to the classification of music. In [37] the authors distinguish between instruments (e.g. Brass, Keyboard, and String) by extracting features such as ZCR, Short Time Energy (STE), Bandwidth, Pitch, Formant Frequencies and Mel-Frequency Cepstral Coefficients (MFCC). These features are computed from short frames of the audio signal. The mean and standard deviations of the features over all frames add up to the feature vector that represents the signal. Classification is performed by GMM and NN. Instrument recognition is proposed in [43]. The authors extract Pitch, Onset Asynchrony, and information about Tremolo and Vibrato of the audio sample. The Fisher projection method is used to build a hierarchical Fisher classifier. Music genre classification is addressed in [23]. In this paper the authors propose the Discrete Wavelet Packet Decomposition Transform to distinguish music genres.

Structural music analysis tries to extract similarities and recurrences in a piece of music. A comprehensive structural analysis is performed in [41]. Autocorrelation is computed to extract Rhythm from the wavelet-decomposed

signal. Pitch Class Profiles in combination with HMM separate chords. Vocal and instrumental sections are characterized in terms of Octave Scaled Cepstral Coefficients (OSCC). An SVM trained with OSCC features separates vocal from instrumental sections.

Environmental sound recognition addresses the identification of sounds that do not originate from speech or music. The range of environmental sounds is extremely wide. Hence, most investigations concentrate on a restricted domain. A popular research field is audio recognition in broadcasted video. In [38] the authors recognize the scene content of TV programs (e.g. weather reports, advertisement, basketball and football games) by analyzing the audio track of the video. They extract Pitch, Volume Distribution, Frequency Centroid and Bandwidth to characterize TV programs. Classification is performed by a neural network for each class. A well investigated problem is highlight detection in sport videos. The authors of [57] retrieve crucial scenes in soccer games by analyzing play-breaks. Whistles, that often refer to play-breaks in sports, are detected using Spectral Energy within an appropriate frequency band. Another indicator for highlights is the audience. Excitement is quantified by Loudness, Silence, and Pitch. A similar approach is followed by Xu [65]. The authors analyze keywords in commentator speech and audience which are relevant to important actions of the game. They apply an HMM trained with low level features (Energy and MFCCs including delta and double delta features) to recognize the keywords. Investigations presented in paper [51] address extraction of highlights in baseball games. Beside visual features, the authors extract audio features (e.g. MFCC, Pitch, Entropy). An SVM detects excitement of the audience. Template matching is applied for baseball hit detection. These two audio cues are combined to improve quality of highlight detection. Another area of interest is surveillance and intruder detection. The authors of [9] detect intruders in a room by monitoring variations in a room-specific transfer function. A broad survey of audio features and classification techniques, in context of automatic surveillance is given in [13].

In [67] multilevel classification is proposed. First the authors apply a coarse level segmentation to separate speech, music and environmental sound. In a second step HMMs are employed to analyze environmental

sounds (e.g. footstep, laughter, rain, windstorm). The authors of [32] present an audio indexing system using MPEG-7 features [42]. They apply Audio Spectrum Basis (ASB) and Audio Spectrum Projection (ASP) descriptors to distinguish classes such as "Dog," "Bell," "Water," and "Baby" with HMMs. They show that MPEG-7 descriptors perform similar to MFCC. SVMs are successfully applied to environmental sound recognition in [24]. The authors compare and combine cepstral features (MFCCs) with perceptual features (Total Spectrum Power, Subband Powers, Brightness, Bandwidth, and Pitch). In [24], perceptual features outperform cepstral features. Best results are obtained by a combination of both. Also in [24], SVM performs better than NN and K-NN.

A challenging area of environmental sound recognition is life logging. This research field is concerned with continuously analyzing the environmental sounds surrounding a user. From this information a diary is constructed where major events and the user's activities are stored. Fundamental research in the domain of life logging is performed in the *Forget-me-not* system [36]. *Forget-me-not* is a mobile application that analyzes the activities of a user in his office. This includes monitoring the workstation, telephone, printer and the location of the user. In [1], Aizawa presents a life logging system that captures video and audio. Audio information is considered to detect human voice to recognize conversation scenes. The system supports GPS and provides inertial trackers to measure motion. Additionally it has access to documents, web pages, and emails.

The applications discussed in this section prove the importance of environmental sound recognition for future information systems. Due to this survey, a number of features and classification techniques are identified that are applicable to environments sound recognition.

6 Conclusions and Future Work

More and more research is performed in the domain of environmental sounds. This development is among other reasons driven by the will to automate surveillance and annotation of audio visual media. This thesis addresses the applicability of state-of-the-art audio features for this specific domain. A database containing 617 environmental sounds from five classes was constructed for testing. Experiments show that popular features employed in speech recognition such as LPC coefficients and MFCCs separate classes of environmental sounds well. Furthermore, one may observe that low complex features such as Fourier coefficients and Wavelet coefficients perform poorly on environmental sounds.

The author introduced a set of novel time-based audio features that are easy to compute. They follow an intuitive way to describe the characteristic shape of a waveform. They perform in a satisfactory but far from optimal way. A combination of state-of-the-art features with the introduced novel feature set enables successful classification of more than 85% of the environmental sounds contained in the database.

Three popular classifiers are employed in the experiments. The SVM and the K-NN classifier perform equally well. Both achieve satisfactory precision and recall values. LVQ is more sensitive to the feature data than the other classifiers in the test. LVQ yields satisfactory results for well discriminating features, while its results for weak features are poor.

The results of the investigation are promising for future research in this area. Frame-based analysis may further improve results of file-based classification. While file-based classification operates on entire files, frame-based techniques deal with analysis and classification of short frames of a signal. That involves the neighboring frames for classification of a frame. For this purpose context sensitive classifiers such as Hidden Markov Models and Artificial Neural Networks will be employed. [54]. Further work will include comparison of features discussed in this thesis with MPEG-7 features in the domain of environmental sounds [42].

Another future goal is the distinction of different sounds from the same class. Such a tool may be useful in surveillance, for example identification

by characteristic sounds of footsteps. Besides, focus has to be directed towards the design of new audio features for environmental sounds. These new features should be low-dimensional and easy to compute. Another field of interest are mobile information systems such as life logging and supportive systems for handicapped people.

Eventually, retrieval quality may be improved by employing hierarchical classification. This technique operates on a hierarchy of classes and applies appropriate classifiers for different groups of classes. Visual information usually accompanies audio information. Multimodal retrieval combines information from different media. This approach is one of the most promising directions in multimedia information retrieval.

List of Tables

List of Figures

Appendix

In the appendix the author provides the source code of features used for environmental sound retrieval.

A Implementation

This section contains Java implementations of features employed in the investigations. In Subsection A.1 the Java source code of the Amplitude Descriptor introduced in [45] is presented. In Subsections A.2 and A.3 the author provides source code for Short-Time Energy and Zero Crossing Rate (see Section 3).

A.1 Amplitude Descriptor - LoHAS, LoLAS, AHA

```
package org.vizir.audio.feature;

/**
 *
 * Implementations of features LoHAS (Length of High Amplitude Sequence),
 * LoLAS (Length of Low Amplitude Sequence) and AHA (Area of High Amplitude).
 * The features were introduced in:
 *      Discrimination and Retrieval of Animal Sounds,
 *      Vienna University of Technology
 *      TR-188-2-2005-05
 *      Mitrovic, D. and Zeppelzauer, M.
 *      2005.
 *
 * After construction of the class, the get-functions may be used to retrieve
 * statistical properties such as mean, variance, and median of LoHAS, LoLAS and AHA
 *
 * (c) by Dalibor Mitrovic and Matthias Zeppelzauer
 */

import org.vizir.util.*;
import java.util.ArrayList;


public class AmplitudeDescriptor {

    private float[] mSignal = null;
    private float[] mLoHAS  = null;
    private float[] mLoLAS  = null;
    private float     mAHA    = 0.0f;
```

```
/**
 * Constructs a new Amplitude Descriptor and computes LoHAS, LoLAS and AHA
 *
 * @param signal the input signal
 */
public AmplitudeDescriptor(float[] signal)
{
    this.mSignal = signal;
    mLoHAS = new float[3];
    mLoLAS = new float[3];

    //calculate absolute values of signal
    for (int i=0; i<this.mSignal.length; i++) {
        this.mSignal[i] = Math.abs(this.mSignal[i]);

    }

    //calculate adaptive threshold
    float threshold =
       Statistics.mean(mSignal)+(float)Math.sqrt(Statistics.variance(mSignal));

    //compute LoHAS, LoLAS, and AHA
    boolean new_LoHAS = false;
    boolean new_LoLAS = false;
    int counter_LAS = 0;
    int counter_HAS = 0;
    float accumulator_AHA = 0.0f;

    ArrayList list_LoHAS = new ArrayList();
    ArrayList list_AHA = new ArrayList();
    ArrayList list_LoLAS = new ArrayList();

    for (int i=0; i<this.mSignal.length; i++) {
        if (this.mSignal[i] >= threshold && new_LoHAS) {
            counter_HAS = counter_HAS + 1; //continue HAS
            accumulator_AHA =
             accumulator_AHA + (this.mSignal[i]-threshold); //increase AHA

        }
        else if (this.mSignal[i] >= threshold && !new_LoHAS) {
            //    new HAS
            new_LoHAS = true;
            counter_HAS = 1;
            //    end LAS
            new_LoLAS = false;
            list_LoLAS.add(new Integer(counter_LAS));
            //    init AHA
```

```
            accumulator_AHA = this.mSignal[i]-threshold;
        }
        else if (this.mSignal[i] < threshold && new_LoLAS)  {
            //    continue with LAS
            counter_LAS = counter_LAS+1;
        }
        else if (this.mSignal[i] < threshold && !new_LoLAS) {
            if (new_LoHAS) {
                // end HAS
                list_LoHAS.add(new Integer(counter_HAS));
                new_LoHAS = false;
                // end AHA
                list_AHA.add(new Float(accumulator_AHA));
            }
            //    new LAS
            new_LoLAS = true;
            counter_LAS = 1;
        }

    }

    //copy ArrayLists to float arrays:
    float[] array_LoHAS = convertIntegerListToFloatArray(list_LoHAS);
    float[] array_LoLAS = convertIntegerListToFloatArray(list_LoLAS);
    float[] array_AHA   = convertFloatListToFloatArray(list_AHA);


    //calculate statistical properties from the float arrays:
    this.mLoHAS[0] = Statistics.mean(array_LoHAS);
    this.mLoHAS[1] = Statistics.variance(array_LoHAS);
    this.mLoHAS[2] = Statistics.median(array_LoHAS);

    this.mLoLAS[0] = Statistics.mean(array_LoLAS);
    this.mLoLAS[1] = Statistics.variance(array_LoLAS);
    this.mLoLAS[2] = Statistics.median(array_LoLAS);

    this.mAHA        = Statistics.mean(array_AHA);

}

private float[] convertFloatListToFloatArray(ArrayList list) {
    float[] array = new float[list.size()];
    for (int j=0; j < list.size(); j++) {
        array[j] = ((Float)list.get(j)).floatValue();
    }
    return array;
}
```

```
    private float[] convertIntegerListToFloatArray(ArrayList list) {
        float[] array = new float[list.size()];
        for (int j=0; j < list.size(); j++) {
            array[j] = ((Integer)list.get(j)).floatValue();
        }
        return array;
    }

    /**
     * getLoHAS returns the statistical properties of LoHAS.
     *
     * @return an array with the mean (position [0]),
     * variance (position [1]) and median (position [2]) of LoHAS
     */
    public float[] getLoHAS() {
        return mLoHAS;
    }

    /**
     * getLoLAS returns the statistical properties of LoLAS.
     *
     * @return @return an array with the mean (position [0]),
     * variance (position [1]) and median (position [2]) of LoLAS
     */
    public float[] getLoLAS() {
        return mLoLAS;
    }

    /**
     * getLoHAS returns the mean of AHA.
     *
     * @return the mean of AHA
     */
    public float getAHA() {
        return mAHA;
    }
}
```

A.1.1 Statistical Utility Functions

The implementation of the features of the Amplitude Descriptor (LoHAS, LoLAS, and AHA) requires the calculation of statistical moments of first and second order (mean and variance). Additionally a function that estimates the median is needed. The following class provides implementations for these functions.

```
package org.vizir.util;

/**
 *
 * Utility class to calculate mean, variance and median of an array of float values
 */
public class Statistics {

    /**
     * Compute the mean of the input array
     * @param values an array of float values
     * @return the mean of the input values
     */
    public static float mean(float[] values) {
        float sum = 0.0f;
        for (int i=0; i<values.length; i++) {
            sum += values[i];
        }
        if (values.length > 0)
            return sum/values.length;
        else
            return 0;
    }

    /**
     * Compute the variance of the input array
     * @param values an array of float values
     * @return the variance of the input values
     */
    public static float variance(float[] values) {
        float meanValue = mean(values);
        float[] helper = new float[values.length];
        for (int i=0; i<values.length; i++) {
            // Y = (X-mu)^2
            helper[i] = (values[i] - meanValue)*(values[i] - meanValue);
        }

        float variance = mean(helper);
```

```
        return variance;
    }

    /**
     * Determine the median of the input array
     * @param values an array of float values (unsorted)
     * @return the median of the input values
     */
    public static float median(float[] values) {
        float[] sortedValues = sort(values);
        float med = 0.0f;


        if (sortedValues.length > 0) {
            int halfLen = (int)(sortedValues.length/2);
            if (sortedValues.length % 2 == 0)  { // even length
                med = (float)(0.5 * (sortedValues[halfLen-1] +
                                            sortedValues[halfLen]));
            }
            else { //odd length
                med = sortedValues[(int)((sortedValues.length+1)/2-1)];
            }
        }
        return med;
    }

    /**
     * Simple sort algorithm in O(N^2)
     * @param an array of float values (unsorted)
     * @return the sorted input array array
     */
    public static float[] sort(float[] values) {
        float helper = 0.0f;
        for (int i=0; i<values.length; i++) {
            for (int j=0; j<values.length-1-i; j++) {
                if (values[j] > values[j+1]) {
                    helper = values[j];
                    values[j] = values[j+1];
                    values[j+1] = helper;
                }
            }
        }
        return values;
    }
}
```

A.2 Short-Time Energy

```
package org.vizir.audio.feature;

/**
 *
 * Calculates the short-time energy of a framed audio signal
 */
public class ShortTimeEnergy {

    /**
     * getShortTimeEnergy returns the a float array containing
     * the shorttime-energy for each frame
     *
     * @param signal the input signal
     * @param samplingRate the samplingrate of the input signal
     * @param frameSize    the desired framesize in ms (milliseconds)
     * @return the short time energy per frame
     */
    public static float[] getShortTimeEnergy(float[] signal, float samplingRate,
                                                                float frameSize) {
        int samplesPerFrame = (int) Math.floor(samplingRate / 1000.0 * frameSize);
        int numOfFrames = signal.length / samplesPerFrame;
        float[] ste = new float[numOfFrames];

        for(int j = 0; j < numOfFrames; j++) {
            for(int i = 0; i < samplesPerFrame; i++) {
                try {
                    ste[j] += (Math.pow((signal[j * samplesPerFrame + i]), 2)
                                                                / samplesPerFrame);
                }
                catch (ArrayIndexOutOfBoundsException ex) {
                    ste[j] = -1;
                }
            }
        }

        return ste;
    }
}
```

A.3 Zero Crossing Rate

```
/**
 *
 * Calculates the number of zero crossings in an audio signal
 */
public class ZeroCrossings {

    /**
     * getZeroCrossings calculates the zero crossings per second
     * of the input <code>signal</code>. This is a measure for the
     * fundamental frequency
     * @param signal the input signal
     * @param samplingFrequ the sampling frequency of the input signal
     * @return the number of zero crossings per second
     */
    public static float getZeroCrossings(float[] signal, float samplingFrequ) {
        int numOfZeroCrossings = 0;
        int len = 0, idx = 0;
        float a = 0, b = 0;
        float factor = 0;

        factor = samplingFrequ / (float) signal.length;

        for(int i = 0; i < (signal.length - 1); i++) {
            idx = i + 1;
            a = Math.signum(signal[i]);
            b = Math.signum(signal[idx]);
            if ( a != b) numOfZeroCrossings += 1;
        }
        return numOfZeroCrossings * factor;
    }
}
```

References

[1] K. Aizawa. Digitizing personal experiences: Capture and retrieval of life log. *In Proceedings of the 11th International Multimedia Modelling Conference*, 00:10–15, 2005.

[2] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[3] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on pure and applied mathematics*, 44:141–183, 1991.

[4] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Fifth Annual Workshop on Computational Learing Theory*, pages 144–152, 1992.

[5] M. Brookes. Voicebox is a matlab toolbox for speech processing. *http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html*, 2005.

[6] J. Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89:425–434, 1991.

[7] M. Carey, E. Parris, and H. Lloyd-Thomas. A comparison of features for speech, music discrimination. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, 1:149–152, 1999.

[8] C. Chiu, Y. Chang, and Y. Lai. Analysis and recognition of human vocal emotions. *In Proceedings of the International Computer Symposium*, 1994.

[9] Y. Choi, K. Kim, J. Jung, S. Chun, and K. Park. Acoustic intruder detection system for home security. *In IEEE Transactions on Consumer Electronics*, 51:130–138, 2005.

[10] J. Cooley and J. Tukey. An algorithm for machine calculation of complex fourier series. *Math. Comp.*, 19:297–301, 1965.

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[12] T. Cover and P. Hart. Nearest neighbor pattern classifications. *IEEE transaction on information theory*, 13:21–27, 1967.

[13] M. Cowling. Non-speech environmental sound classification system for autonomous surveillance. *PhD Thesis*, Griffith University, Queensland, Australia, 2004.

[14] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36:961–1005, 1990.

[15] R. Duda, P. Hart, and D. Stork. *Pattern Classification 2nd edition*. Wiley, 2001.

[16] D. Ellis. Matlab audio processing examples. *www.ee.columbia.edu /∼dpwe/resources/matlab/*, 2005.

[17] S. Esmaili, S. Krishnan, and K. Raahemifar. Content based audio classification and retrieval using joint time-frequency analysis. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, 5:665–668, 2004.

[18] B. Feiten and S. Gunzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18:53–65, 1994.

[19] M. Filickner, H. Sawhney, W. Niblack, J. Ashley, W. Huang, B. Dom, M. Gorkani, J. Hafine, D. Lee, D. Petkovic, D. Steel, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer Society Press*, 28:23–32, 1995.

[20] H. Fletcher, E. Blackman, and R. Stratton. Quality of piano tones. *Journal of Acoustical Society of America*, 34:1535 – 1544, 1962.

[21] J. Foote. Content-based retrieval of music and audio. *In Proceedings of the SPIE conference on Multimedia Storage and Archiving Systems II*, 3229:138–147, 1997.

[22] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming: musical information retrieval in an audio database. *Proceedings of the third ACM international conference on Multimedia*, pages 231–236, 1995.

[23] M. Grimaldi, P. Cunningham, and A. Kokaram. A wavelet packet representation of audio signals for music genre classification using different ensemble and feature selection techniques. *In Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 102–108, 2003.

[24] G. Guo and Z. Li. Content-based classification and retrieval by support vector machines. *In IEEE Transactions on Neural Networks*, 14:209–215, 2003.

[25] J. Hadamard. *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton University Bulletin, 1902.

[26] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

[27] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2:578–589, 1994.

[28] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, 1999.

[29] R. Jang. Data clustering and pattern recognition toolbox. *http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/DCPR/*, 2005.

[30] H. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200, 1958.

[31] B. Kedem. Spectral analysis and discrimination by zero-crossings. *IEEE Proceedings*, 74:1477–1493, 1986.

[32] H. Kim, N. Moreau, and T. Sikora. Audio classification based on MPEG-7 spectral basis representations. *In IEEE Transactions on Circuits and Systems for Video Technology*, 14:716–725, 2004.

[33] T. Kohonen. Improved versions of learning vector quantization. *Proceedings of the International Joint Conference on Neural Networks (IJCNN '90)*, 1:545–550, 1990.

[34] T. Kohonen. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[35] T. Kohonen. Learning vector quantization. *The handbook of brain theory and neural networks*, pages 537–540, 1998.

[36] M. Lamming and M. Flynn. 'forget-me-not' intimate computing in support of human memory. *In Proceedings of FRIEND21 International Symposium on Next Generation Human Interface*, pages 125–128, 1994.

[37] M. Liu and C. Wan. Feature selection for automatic classification of musical instrument sounds. *In Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 247–248, 2001.

[38] Z. Liu, J. Huang, Y. Wang, and T. Chuan. Audio feature extraction and analysis for scene classification. *In IEEE Workshop on Multimedia Signal Processing*, 20:343–348, 1997.

[39] J. Loehlin. *Latent variable models: An Introduction to Factor, Path, and Structural Analysis (3rd edition)*. Lawrence Erlbaum Assoc., 2001.

[40] J. Ma, Y. Zhao, and S. Ahalt. Osu svm classifier matlab toolbox. *http://www.ece.osu.edu/∼maj/osu_svm/*, 2005.

[41] N. Maddage, C. Xu, M. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. *In Proceedings of the 12th annual ACM international conference on Multimedia*, pages 112–119, 2004.

[42] B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7*. Wiley, 2002.

[43] K. Martin and Y. Kin. Musical instrument identification: A pattern-recognition approach. *In Proceedings of the 136th meeting of the Acoustical Society of America (ASA)*, 1998.

[44] Mathworks. Matlab. *http://www.mathworks.com*, 2005.

[45] D. Mitrovic and M. Zeppelzauer. Discrimination and retrieval of animal sounds. *In Proceedings of the IEEE Conference on Multimedia Modelling 2006 (accepted)*, 2006.

[46] Y. Muthusamy, E. Barnard, and R. Cole. Reviewing automatic language recognition. *In IEEE Signal Processing Magazine*, 11:33–41, 1994.

[47] E. Pampalk. A matlab toolbox to compute similarity from audio. *In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 254–257, 2004.

[48] L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[49] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Prentice Hall Inc., Englewood Cliffs, New Jersey, 1978.

[50] D. Reynolds and R. Rose. Robust text-independent speaker identification using gaussianmixture speaker models. *In IEEE Transactions in Speech and Audio Processing*, 3:72–83, 1995.

[51] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs. *in Proceedings of the ACM International Conference on Multimedia*, pages 105–115, 2000.

[52] G. Salton and M. McGill. *Introduction to modern information retrieval*. New York [etc.] : McGraw-Hill, 1983.

[53] E. Schreirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, 2:1331–1334, 1997.

[54] K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden Markov model structure for information extraction. *In Papers from the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

[55] SPSS. Spss. *http://www.spss.com/*, 2005.

[56] X. Sun. Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, 2002.

[57] D. Tjondronegoro, Y. Chen, and B. Pham. Applications ii: The power of play-break for automatic detection and browsing of self-consumable sport video highlights. *In Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 267–274, 2004.

[58] T. Tremain. The government standard linear predictive coding algorithm: Lpc-10. *In Speech Technology Magazine*, 1:40–49, 1982.

[59] J. Tukey, B. Bogert, and M. Healy. The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe-cracking. *In Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed.)*, pages 209–243, 1963.

[60] C. van Rijsbergen. *Information Retrieval*. http://www.dcs.gla.ac.uk/Keith/Preface.html, 1979.

[61] V. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 1999.

[62] W. Watanabe. *Pattern Recognition: Human and mechanical*. Wiley, 1985.

[63] Wikipedia. Inverse problem. *http://en.wikipedia.org/wiki/Inverse_problem*, 2005.

[64] T. Wold, D. Blum, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3:27–36, 1996.

[65] M. Xu, L. Duan, L. Chia, and C. Xu. Audio keyword generation for sports video analysis. *In Proceedings of the 12th annual ACM international conference on Multimedia*, pages 758–759, 2004.

[66] M. Zeppelzauer. Discrimination and retrieval of animal sounds. *Technical Report TR-188-2-2005-06, http://www.ims.tuwien.ac.at/publication_master.php*, 2005.

[67] T. Zhang and C. Kuo. Hierarchical classification of audio data for archiving and retrieving. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, 6:3001–3004, 1999.

[68] T. Zhang and C. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *In IEEE Transactions on Speech and Audio Processing*, 9:441–457, 2001.