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 Animal Sounds

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

unter Anleitung von
ao. Univ. Prof. Mag. Dr. Horst Eidenberger
Univ. Prof. Dipl. Ing. Dr. Christian Breiteneder

durch

Matthias Zeppelzauer
Matr. Nr. 9926063

Wildenhaggasse 12
A-3423 St. Andrä Wördern

Wien am 15. Oktober 2005

Datum

Unterschrift

Abstract

Hearing is the second most important human sense after vision. Research primarily concentrated on visual information retrieval in the past. Only few research took place in the area of audio information retrieval. The main focus was speech recognition for a long time. Recently, information retrieval of music gained importance through the distribution of digital music over the internet. Most sounds in our environment are neither speech nor music. Environmental sounds contain important information we permanently use for orientation. Environmental sound recognition is an upcoming research area that enables a variety of new applications, such as life logging and automatic surveillance. So far, few research has been performed in the area of animal sound retrieval.

In this thesis, the author identifies state-of-the-art techniques in general purpose sound recognition by a broad survey of literature. Based on the findings, this thesis gives a thorough investigation of audio features and classifiers and their applicability in the domain of animal sounds. Techniques developed especially for environmental sound recognition rarely exist. Therefore techniques from other areas of audio processing are employed. Multiple features originally developed for speech recognition are applicable to environmental respectively animal sounds as well. Furthermore, music information retrieval methods are employed. Classification is performed by popular machine learning techniques.

Due to the lack of publicly available data, a large database of animal sounds is built. Additionally, the author introduces a set of novel audio descriptors. The features are time-based and characterize properties of an audio waveform that are significant for human perception. Their quality is compared with that other popular audio features. Experiments show that the new descriptors perform comparably to state-of-the-art features. The results of animal sound retrieval are encouraging and motivate further research in this domain.

Zusammenfassung

Das Hören ist der zweit wichtigste menschliche Sinn nach dem Sehen. Forscher konzentrierten sich in der Vergangenheit hauptsächlich auf visual information retrieval. Bisher fand nur wenig Forschung im Bereich des audio information retrievals statt. Der Schwerpunkt lag lange Zeit auf der Erkennung von Spracher. In letzter Zeit gewann music information retrieval an Bedeutung durch die Verbreitung digitaler Musik über das Internet. Die meisten Geräusche in unserer Umgebung sind weder Sprache noch Musik. Umgebungsgeräusche enthalten wichtige Informationen, die wir ständig zur Orientierung verwenden. Die Erkennung von Umgebungsgeräuschen ist ein aufstrebendes Forschungsgebiet, das neue Anwendungen wie etwa life logging und automatisierte Überwachung ermöglicht. Bisher wurde wenig Forschung im Bereich der Erkennung von Tiergeräuschen betrieben.

In dieser Magisterarbeit stellt der Author den aktuellen Stand der Technik in der Geräuscherkennung durch eine umfassende Literaturstudie dar. Auf Grundlage der gewonnenen Erkenntnisse präsentiert diese Arbeit eine gründliche Untersuchung von Audio-Merkmalen und Klassifikatoren und prüft deren Andwendbarkeit im Bereich der Tiergeräusche. Es existieren kaum Techniken, die speziell für die Erkennung von Umgebungsgeräuschen entwickelt wurden. Daher werden Techniken aus anderen Bereichen der Audioverarbeitung herangezogen. Viele Merkmale, die ursprünglich für Spracherkennung entwickelt wurden, sind auf Umgebungs- bzw. Tiergeräusche anwendbar. Weiters werden Methoden aus dem music information retrieval verwendet. Zur Klassifikation kommen Techniken aus dem Bereich des machine learning zum Einsatz.

Da öffentlich nutzbaren Daten fehlen, wurde eine Datenbank von Tiergeräuschen aufgebaut. Weiters wird ein Satz von neuen Audio-Deskriptoren vorgestellt. Die Merkmale sind zeitbasiert und beschreiben Eigenschaften eines audio Signals, die für die menschliche Wahrnehmung maßgeblich sind. Die Qualität der Deskriptoren wird mit jener von anderen beliebten Audio-Merkmalen verglichen. Experimente zeigen, dass die neuen Deskriptoren vergleichbare Leistungen bringen wie aktuelle Merkmale. Die Ergebnisse für animal sound retrieval sind vielversprechend und motivieren weitere Forschung in diesem Bereich.

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 "coffee queen" Divotkey and Ms. Karyn Laudisi. Furthermore, I want to thank my supervisors ao. Univ. Prof. Mag. Dr. Horst "Mr. Iterative Process" Eidenberger and Univ. Prof. Dipl. Ing. Dr. Christian Breiteneder.

Contents

List of Tables

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 focused 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. Audio retrieval is employed in multimodal information retrieval, where visual, textual, and acoustic information is combined to take advantage of synergetic effects. Audio recognition is also applied for automatic extraction of semantic annotations in multimedia databases.

This thesis concerns with the retrieval of animal sounds. Animal sounds are a subset of environmental sounds. The investigation surveys a broad set of audio features and several classifiers. Additionally, the author introduces a new set of features and evaluates their quality by a selected set of classes of animal sounds. Due to the complexity and the nearly infinite domain of non-speech sounds, the quality of recognition is typically lower than the quality of speech recognition, which is already well understood. Retrieval results presented in this thesis for the domain of animal sounds are comparable to that of state-of-the-art research in the area of environmental sound recognition.

1.1 Motivation & 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. 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 every year, 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 for retrieval systems. Early approaches employed query-by-example techniques. Later, query-by-humming gained importance especially in the field of music retrieval [23]. A retrieval task is always a tradeoff between universality and assumptions - about the domain, about the media, and about the user.

The problem of content-based audio retrieval can be stated as follows: Content-based audio retrieval concerns with searching in multimedia databases for audio samples specified by a query that describes properties of the desired audio samples. In general, 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 scope of the defined classes. Based on the parametric model, retrieval is performed by defining and evaluating a query.

1.2 Contribution

Animal sounds are a domain of environmental sounds that has not been investigated in detail yet. Some investigations consider animal sounds among other classes of sound [30], [22]. These investigations concern with classes of environmental sounds in general. To the authors' knowledge there is no prior work analyzing the discrimination of animal sounds from each other. Animal sound retrieval is a new domain of research. Hence, there are no

investigations the results in this thesis can be compared with directly. Furthermore, there is no official and commonly accepted reference database for animal sounds.

This thesis addresses the identification of an efficient method for automatically distinguishing between sounds of different animals. The 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 animal sound recognition. Therefore a database consisting of several hundred animal sounds is built. Traditional features developed for speech recognition and features applied in audio segmentation and music retrieval are compared. Additionally, the author introduces a set of novel features and compares their performance with popular audio features. The introduced 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. Besides, an extensive survey of state-of-the-art features and classifiers is presented. Additionally, a comprehensive overview of related research in the field of content-based audio retrieval is given.

1.3 Applications

Animal 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 is equipped with a microphone and a mobile device that is responsible for retrieval. The user is 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. Animal sound retrieval can be applied, for example to recognize barks of a watchdog that often signalize crucial events.

A traditional field of research is the annotation of time-dependent media. Animal sound retrieval may be part of a system that automatically generates meta-information from audio and video streams. A related application is the annotation of movies in a multimedia database to improve search capabilities.

Additionally, life logging could take advantage of such a technique. A life log accompanies human users during their working life and leisure time and automatically captures and annotates events of interest in a multimodal electronic 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 signals and combined with data of other sensors. The resulting diary consists of retrieved annotations associated with a time stamp. Animal sound retrieval may be useful in a life logging application, imagine a visit to the zoo. A thorough survey of applications related to content-based audio retrieval and animal sound retrieval is given in Section 5.

1.4 Organization

The remainder of the thesis is organized as follows: In Section 2 the principles of pattern recognition, information 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 the basic ideas of audio retrieval are discussed in general. First, the field of information retrieval is surveyed. Then the author presents the fundamentals of pattern recognition and finally, basics of digital audio are discussed.

2.1 Information Retrieval

Information retrieval concerns with searching documents in a database by a textual query. Early applications basically 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 [44] and Rijsbergen [50].

In the last decades the number of available media has grown. Audio and video 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 to retrieve audio and video data. Manual creation of textual metadata from multimedia objects by humans is not applicable, because it is too time consuming and error prone. 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 concerned, content-based image retrieval (CBIR), content-based video retrieval (CBVR) and content-based audio retrieval (CBAR) are distinguished.

2.2 Pattern Recognition

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 task consists of multiple parts. A sensor (e.g. a microphone or video camera) provides the system with the raw data of a signal. The size of the data is reduced by feature extraction. This results in a more abstract description that represents the most meaningful information that characterizes the signal well. Based on this representation, classification is performed. Classification is a process that groups similar patterns, represented by features together. In a content-based retrieval 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 dogs " or "find pictures of cars") can be formulated as query. In the following the task of pattern recognition is considered in more detail.

According to Watanabe, patterns are "the opposite of chaos" [52]. 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 discrete Fourier Transform (DFT) 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. Section 3.4 gives a thorough discussion of a variety of audio features. The author presents mathematical foundations and describes details concerning the application of the features in content-based audio retrieval.

As mentioned before, 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 the existing features.

The objective of classification is to predict the class membership of a pattern represented by a corresponding feature vector. A class ω_i is defined by a class label $i \in N$. Each pattern respectively each 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 output 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. During training the classifier determines the class boundaries based on training vectors contained in the training set. After training, the classifier is fit to the data and ready for classification. The quality of the classifier 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 test and training vectors, but all arbitrary vectors that belong to one of the selected classes. This is the generalization ability of a classifier [15]. In Section 3.5 three classifiers, employed in this thesis, 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 beyond multiple classes. A good representation of data by features is a necessary condition for successful pattern recognition. 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 between. Analogously, classifiers cannot be evaluated in isolation. They have to be considered together with the features they operate on.

Pattern recognition tasks (e.g. remote sensing, computer vision, image understanding, and content-based retrieval) are inversions of well-posed problems. For example, computer graphics is the well-posed inversion of pat-

tern recognition and content-based image retrieval. Similarly sound synthesis is the well-posed inversion of audio recognition respectively content-based audio retrieval. In general, an inverse problem concerns with the estimation of model parameters through the manipulation of observed data [53]. 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 labels (e.g. terms like "car" and "cat", properties like "male" and "female", and concepts like "outdoor" and "indoor").

The semantic gap is related to the ill-posed nature of content-based retrieval. The semantic gap refers the mismatch between high-level concepts and low-level descriptions. In content-based retrieval the semantic gap is placed between the content of media and textual information describing the semantics of the content. The following expresses this circumstance:

> It is annoying trying to retrieve Hollywood kisses in a movie database by color, texture and shape features. On the technical level this fact is called "semantic gap". [16]

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 semantic gap and apply similar techniques to narrow it.

2.3 Content-Based Retrieval Systems

The first content-based retrieval systems came up in the nineties. One of the first image retrieval systems was QBIC [20]. 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 [54]. 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 to the documents, 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 find similar documents in the database by applying a similarity model. Another method is query-by-text, where the user defines the desired class of documents or terms that describe 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 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. A typical example is given in Figure 1. The recall-precision graph shows the recall on the abscissa for different precisions on the ordinate. The recall-precision pairs are obtained by varying the number of retrieved documents.

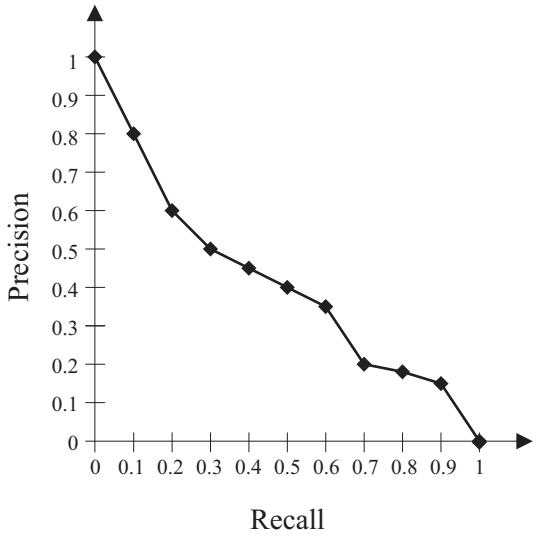


Figure 1: A typical recall-precision graph, illustrating the tradeoff between the two measures.

2.4 Content-Based Audio Retrieval

The rising number of audio, video, and image databases states 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 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 can be reused in CBAR. Today speech recognition is well understood and well engineered. The results of CBAR can currently not compare to 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 concerns with the 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. Speech recognition is extensively surveyed by Rabiner and Juang in [41].

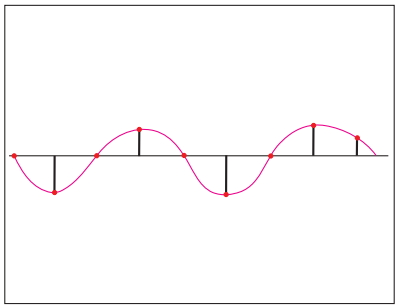
In the last decade analysis and retrieval of music became a popular research field [18]. On the one hand side research deals with retrieval of instruments, artists and musical genres. On the other hand side researchers concentrate on the extraction of semantic information in pieces of music.

Another field of research is environmental sound retrieval which comprises all types of sound that are not speech and music. Since the domain of environmental sounds is arbitrary in size, most investigations confine to a limited domain of sounds. A thorough investigation of related work is given in Section 5.

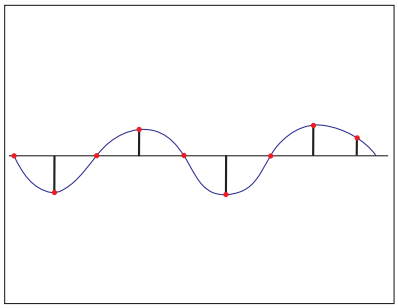
2.5 Digital Audio

Prior to working with sound it is advantageous to become acquainted with its fundamentals. 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. These vibrations generally have frequencies ranging from 20 Hz to 20 000 Hz.

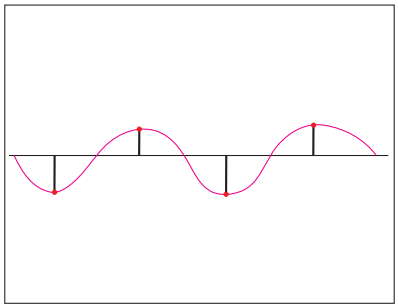
Since physical sound is analog it has to be digitized to be processed with digital hardware. Usually digitalization of sound means recording a number of samples of that sound at certain time intervals. In order to enable a perfect reconstruction of the digital signal, the analog signal has to be sampled 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 2.
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 of 16 Bits. Such an encoding preserves all perceivable frequencies and does not introduce audible quantization noise.



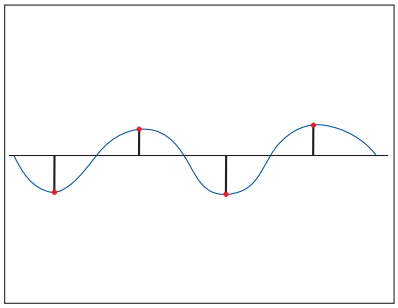
(a) oversampled signal



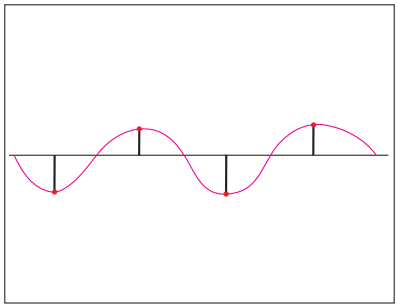
(b) reconstructed oversampled signal



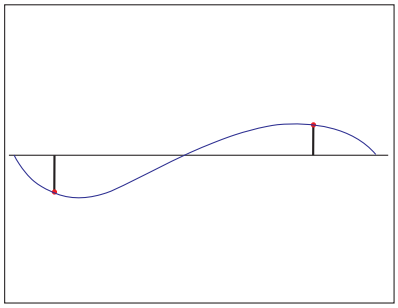
(c) signal sampled with twice its bandwidth (Nyquist-Shannon limit)



(d) reconstructed signal



(e) undersampled signal



(f) reconstructed undersampled signal

Figure 2: 2(a) to 2(d) illustrate, that no gain is achieved through oversampling. The reconstructed signal in 2(b) is identical to the signal that was sampled at twice its bandwidth. 2(e) and 2(f) illustrate the devastating effect of undersampling. The signal cannot be reconstructed properly.

3 Experiments

This thesis examines ways to distinguish between animal sounds. To the authors' knowledge, distinction of animal sounds has not been investigated yet. Animal sound retrieval is a new branch of research in the field of environmental sound recognition. This section presents the experiments performed. Different features and classifiers are applied and compared. First, the scope and the objectives of the experiments are discussed. Then the test setup and the framework that supports the experiments are described. Finally features and classifiers employed in the tests are presented.

3.1 Scope

Four animals, namely birds, cats, cows, and dogs are chosen for the investigation. Sounds by birds and cats respectively by cows and dogs show significant similarity on the spectral domain. By establishing the ground truth of the data set, it becomes apparent that it is even difficult for human observers to correctly classify animals only by their sound. It shows that the perception of certain sounds of cats are similar to sounds of cows. The similarities on the technical and perceptual level qualify the selected classes to measure the performance of features and classifiers without bias.

There is no publicly available reference database of animal sounds. The author built a custom database of sound samples from an internet search. The database contains 383 samples (99 birds, 110 cats, 90 cows, and 84 dogs). The data have a sample rate of 11 025 Hz, are quantized to 16 bit and are single channel. A sound sample contains one or more repeated sounds of an animal (e.g. repeated barks of a dog). Additionally some samples contain background noise of other animals. File lengths and loudness levels vary over the samples. Classification is performed on entire sound samples. Each sound sample is assigned to one of the four classes.

The goal of this thesis is to compare techniques in context of the domain of animal sounds. A system should be developed that is able to correctly classify about 80% of the animal sounds contained in the test set. The system learns the differences characterizing the classes of animal sounds from a training set that is much smaller than the test set. Techniques

applied for retrieval should be easy to compute, to meet the demands of mobile applications.

3.2 Setup

Numerous experiments are preformed 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 and training set are located. Optionally, a file can be specified, where the test and training set are stored as a binary file.
- **feature** specifies the features to compute. One or more features can be declared. 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 technique that is used for classification together with its specific parameters.

Currently, the following outputs are defined:

- **data file** is a binary file where the whole data from the test and training set are stored. That includes the entire samples of all files annotated with metadata such as sample rate, file size, file path, class name and class label.
- **feature 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 arbitrary combinations of features and classifiers. All experiments are conducted in MATLAB using an extensible framework introduced in Section 3.3 that supports experiment files defined as above.

Common to all experiments is the ground-truth. The sample database is split into a test set and a training set. The training set comprises 12 samples per class. The training samples are chosen randomly to gain an unbiased training set. The remaining samples form the test set: 87 bird samples, 98 cat samples, 78 cow samples, and 72 dog samples. The training set is chosen very small to prove the ability to generalization of the classifiers.

The experiments split in two series. In the first run, all features are tested individually with the three selected classifiers. That are k-nearest neighbor (K-NN), learning vector quantization (LVQ), and a support vector machnine (SVM). The results of these experiments are discussed in Section 4.1. In the second run, the features are combined to improve the quality of retrieval. The corresponding results are illustrated in Section 4.2. The large number of experiments enables an objective comparison of the employed classifiers in Section 4.3.

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 mentioned in Section 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 [40], [5], [17], and [35] 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. Due to this it is feasible to represent an experiment, that comprises an entire retrieval process, by a short description 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 them. 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 done by implementing an interface that encapsulates its specific implementation.

The framework provides a mechanism to store and import sample and feature data. This speeds up repeated experiments enormously and allows further analysis of feature data. The structure of the framework is depicted in Figure 3. Experiments are performed on a PC with an Athlon 64 3000+ and 512 MB of RAM. MATLAB version 6.5 is used as programming environment.

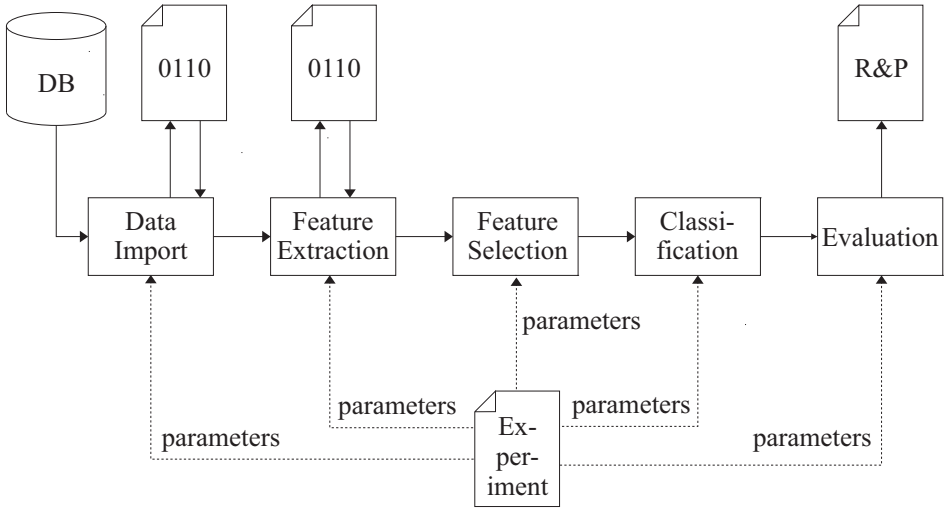


Figure 3: The MATLAB framework employed for the experiments. Each experiment is represented by a configuration file that defines the parameters of the individual retrieval processes, such as feature extraction, features selection, classification and evaluation.

3.4 Feature Extraction

Content-based retrieval usually does not operate on the original data, instead features that represent the content more efficiently, are computed. For illustration consider one second of an audio file in CD-quality: The original data contain 44 100 samples. The first several hundred Mel-Frequency Cepstral Coefficients (MFCCs) of the same signal may suffice for retrieval. This is a crucial reduction of the amount of data that has to be processed.

There is no distinct 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 Transform, Cosine Transform, and Wavelet Transform. Another way to classify audio features is to analyze whether they aim to imitate properties the human sense of hearing. 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 ear is regarded as a superordinate property. Time-based features in the investigation comprise Zero Crossing Rate and Short-Time Energy. Additionally, the author introduces a set of new time-based features that describe the shape 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 concerned 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. Perceptual features are Sone, Pitch, Bark- and Mel-scaled Cepstral Coefficients, PLP, and RASTA-PLP.

In the remainder of 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 animal sounds.

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. It is computed as follows:

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

In the experiments the 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 MFCCs, see Section 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 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. 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 $O(N \cdot \log N)$. 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 7 is commonly referred to as *the DCT*.

$$f_j = \sum_{n=0}^{N-1} s(n) \cdot \cos\left(\frac{j\pi}{N}\left(n + \frac{1}{2}\right)\right) \tag{7}$$

Equation 8 presents the variant DCT-III which is commonly referred to as *the inverse DCT* (IDCT).

$$s_j = \frac{1}{2} f\,(0) + \sum_{k=1}^{N-1} f(k) \cdot \cos\left(\frac{n\pi}{N}\left(j + \frac{1}{2}\right)\right) \tag{8}$$

Similarly to DFT the computation for the DCT is in $O\,(N \cdot \log N)$. In practice DCT is often used for lossy data compression (e.g. JPEG). 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 Components Analysis (PCA), a linear transform that optimally keeps the subspace that has largest variance.

Selected DCT coefficients, especially the low frequency components, are often used as a feature for classification. Analogously to the DFT the first 60 DCT coefficients are used for retrieval in the experiments.

3.4.4 Wavelet Transform

The Wavelet Transform (WT) is a time-frequency transform. 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]. Selection of the optimal mother wavelet depends on the application. Recently, the WT started to replace the FT in several research and application areas, such as signal processing, speech recognition, and astrophysics.

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 9.

$$c\left(a,b\right)=\int_{-\infty}^{\infty}s\left(n\right)\psi\left(an+b\right)\mathrm{d}n \tag{9}$$

CWT is commonly used for signal analysis in scientific research. It is infinitely redundant but sometimes useful to comprehend certain signal properties. The DWT uses a specific subset of scale and translation values which fulfill the conditions in Equation 10.

$$\psi\left(2^{k}n+l\right)\text{ with }k,l\in Z \tag{10}$$

DWT is employed in computer science and engineering as a means of signal coding and compression. The computational complexity of the DWT is in $O\left(N\right)$. In practice the DWT is computed by the use of FIR filters. Similar to DCT and DFT coefficients, DWT coefficients are directly employed as features. In the experiments feature vectors that contain the first 50 DWT coefficients are used. The mother wavelet employed is the Haar wavelet.

3.4.5 Constant Q Transform

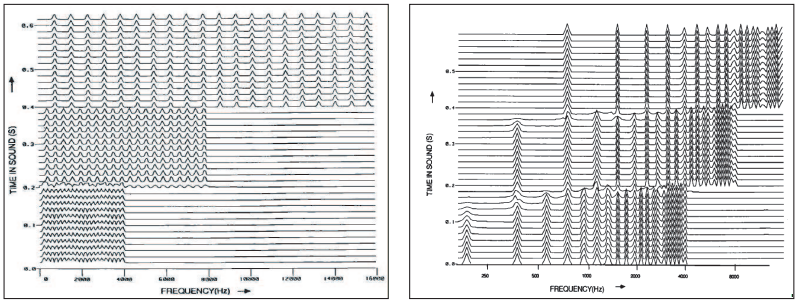
In order to overcome the shortcomings of the Fourier Transform for analysis of Western music, Brown introduced the constant Q Transform (CQT) in [6]. The DFT yields frequency components that are separated by a constant frequency difference and with a constant resolution. These frequency components do 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 11 illustrates the computation of the CQT:

$$X\left(k\right)=\frac{1}{M\left(k\right)}\sum_{n=0}^{M(k)-1}W\left(k,n\right)s\left(n\right)exp\left(\frac{-i2\pi Qn}{M\left(k\right)}\right) \tag{11}$$

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 $Q = \lfloor f/\delta f \rfloor$.

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 4(b) and 4(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 4: Fourier Transform and constant Q Transform 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 cannot be measured physically, since it is an auditory sensation. Two sounds with measurably different frequencies do

not need to have two different pitches but difference in the perceived pitch implies different frequencies. The author employs a pitch detection algorithm devised by Sun in [46]. 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 feature.

For the experiments the MATLAB toolbox of Pampalk is employed [40]. The author computes sone values for 40 frequency bands with a window size of 256 samples. Mean and variance serve as features.

3.4.8 Cepstral Coefficients

Cepstral Coefficients (CCs) are a popular feature in audio retrieval [33], [55]. The authors of [49] 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 similarly computed as 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. Analogously to above 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 [42], [48]. The goal of LPC is to estimate the basic parameters of a speech signal, e.g. pitch, formants, spectra, 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. It is beyond the authors' knowledge that LPC coefficients have been introduced to environmental sound recognition. In [38] the author successfully applies LPC coefficients to environmental sound recognition in the scope 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 on 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 these problems 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 one 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.

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 [17]. 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 of 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.

From 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 any 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.

After band-pass filtering the equal loudness curve and cubic-root 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 all 18 coefficients are selected for retrieval.

3.4.12 Zero Crossing Rate

The Zero Crossing Rate (ZCR) is the number of zero-crossings in time domain within one second. According to Kedem [29] the ZCR is a measure for the dominant frequency in a signal. The mean ZCR for entire sample files is used as 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 12 illustrates the computation.

$$STE = \Delta t \sum_{n=1}^{N} |s\,[n]|^2 \tag{12}$$

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

3.4.14 LoHAS, LoLAS & AHA

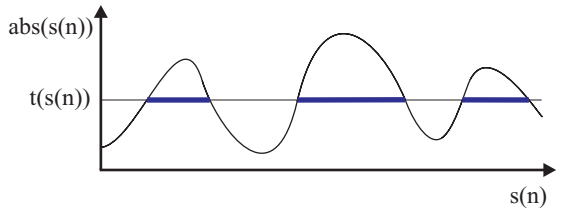
The author introduces a set of new time-based low-level features for audio [38]. The features follow a simple perceptually driven approach. A human observer distinguishes sounds among other things by the distribution of loud portions and silent portions. Sounds often consist of similar recurrent fragments. Animal sounds match this concept very well, for example the tweets of a bird and the barks of a dog are repeating sounds. The human hearing uses this information to distinguish and recognize sounds. For example, the barks of a dog differ from a low of a cow because the repeat rate and the length of the single sounds are different. On the technical level that means that the high energy segments are different in length. Analogously, 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 5(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 5(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 their 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 5(c) illustrates this concept.

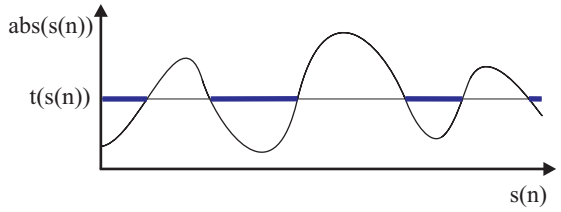
Statistical properties of LoHAS, LoLAS, and AHA are considered to build features that describe entire sample files. The final features comprise mean, standard deviation, and median of LoHAS and LoLAS over the entire signal. Additionally, the mean of AHA is extracted. This results in a seven dimensional feature vector which is used for classification.

3.5 Classification

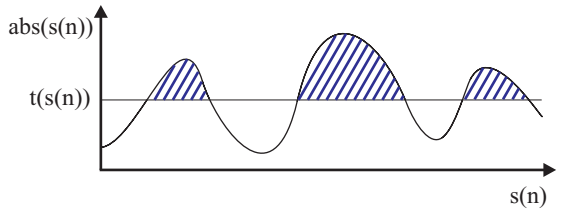
In this section the classifiers employed in the experiments are described. There is a large number of classification techniques following different approaches. Statistical methods such as Bayes classification and Gaussian mixture models try to estimate the probability density function of the underlying data. Another group of classifiers are learning algorithms that



(a) LoHAS



(b) LoLAS



(c) AHA

Figure 5: 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).

employ artificial intelligence techniques. Most algorithms fit a parametric model to the underlying data. There are supervised learning methods such as support vector machines (SVM) and neural networks and non-supervised techniques such as self organizing maps. A classification technique similar to self organizing maps is learning vector quantization (LVQ). Beside parametric techniques (e.g. support vector machines) there are non-parametric techniques such as nearest neighbor.

Three supervised classifiers are selected for the experiments. The simplest way to classify feature vectors is the nearest neighbor rule. We employ a K-NN classifier, which is a generalization of the nearest neighbor classifier.

In the experiments the implementation of Roger Jang is applied [28]. Additionally, the author implements learning vector quantization using standard MATLAB routines. Finally, a SVM is applied for classification with different kernels. For this purpose the OSU SVM MATLAB toolbox is used [35].

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

$$s = \arg\min_i \|\mathbf{x} - \mathbf{x}_i\|, \; 1 \leq i \leq N. \tag{13}$$

Similarity in nearest neighbor classification can be measured by any similarity (distance) measure. Usually Euclidean distance is used. This assignment scheme partitions the feature space according to a Voronoi tessellation. Each cell belongs to a class. Figure 6 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 only 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:

$$i = \arg\max_j k_j, \; 1 \leq i \leq c \tag{14}$$

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 ($O(N)$).

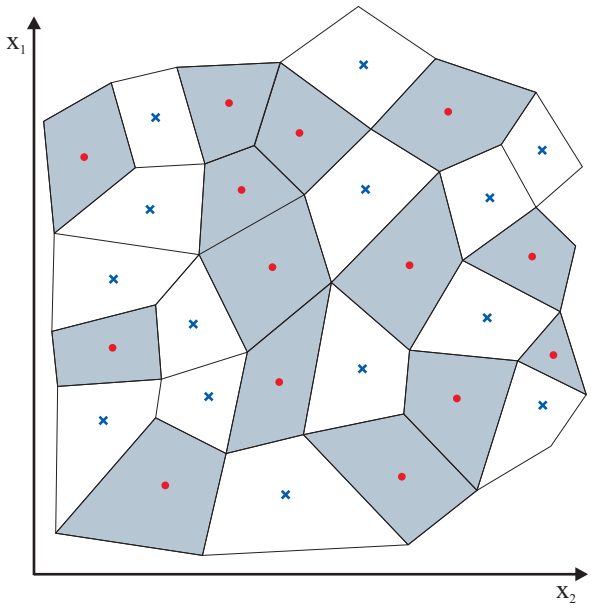


Figure 6: Voronoi tessellation in R^2 of a binary classification problem. Dots are feature vectors of class A, crosses are feature vectors of class B. The gray area is the decision region of class A.

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 test the quality of the features. Features that discriminate classes well, provide disjoint partitions of the feature space. Satisfactory results with the NN algorithm indicate such a partitioning in the feature space.

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 [31] and is related to Self-organizing maps, also by Kohonen [31].

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 assign a subset of codebook vectors to each class

S_k and then search the codebook vector $\mathbf{m}_i$ with 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

$$c = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \tag{15}$$

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 16 to 18:

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \qquad \mathbf{x}, \mathbf{m}_c \in S_k \tag{16}$$

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)], \quad \mathbf{x} \in S_i, \mathbf{m}_c \in S_j,\ i \neq j \tag{17}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t), \qquad i \neq c \tag{18}$$

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 7. The screenshots were made with the LVQ visualization tool developed by Borgelt [3].

The optimized-learning-rate LVQ1 (OLVQ1) is an improved version of the LVQ1 presented above. OLVQ1 differs from LVQ1 in the fact that it uses an individual learning rate $\alpha_i(t)$ that is assigned to each $\mathbf{m}_i$. Let c be defined in Equation 15, and let $f(\mathbf{x}) = +1$, $f(\mathbf{x}) = -1$ denote correct

respectively incorrect classification of $\mathbf{x}$. Equations 19 to 21 define the new process:

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha_c(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right], \quad f(\mathbf{x}) = +1 \tag{19}$$

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha_c(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right], \quad f(\mathbf{x}) = -1 \tag{20}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t), \qquad i \neq c \tag{21}$$

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.).

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.

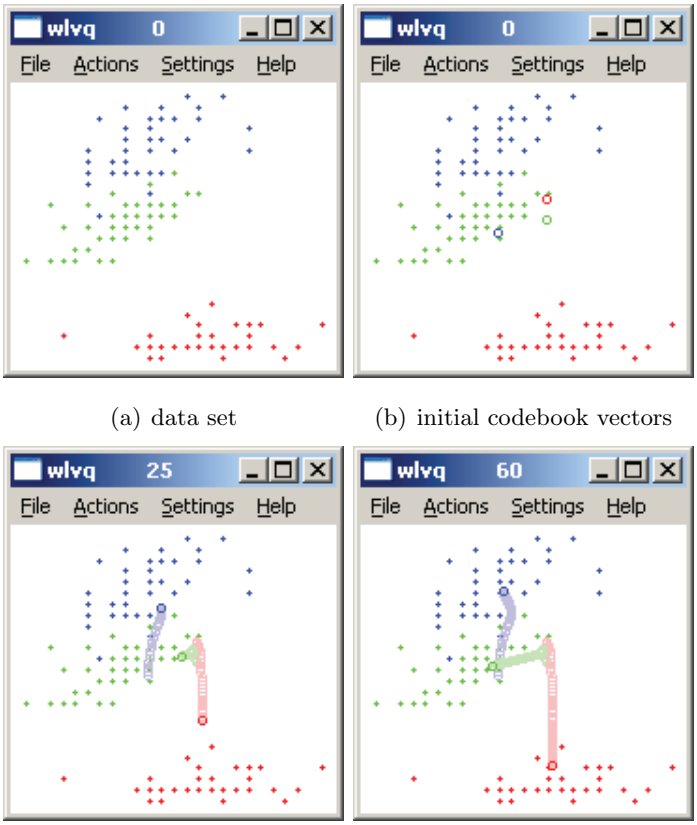
In the experiments an LVQ with 8 hidden neurons, a learning rate of 0.01 and 200 epochs is used. The classifier is supplied with the distribution of classes in the training set.

3.5.3 Support Vector Machines

A popular classifier is the support vector machine (SVM) [4][51]. SVMs are supervised, statistical learning methods applicable for classification and regression. 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 point of each cloud (these points are the support vectors). The distance between the support vectors and the hyperplane is called margin. Figure 8 depicts the difference between a suboptimal and an optimal separating hyperplane. The hyperplane is not constructed in feature space, instead the saddle point of the following Lagrange functional is calculated:

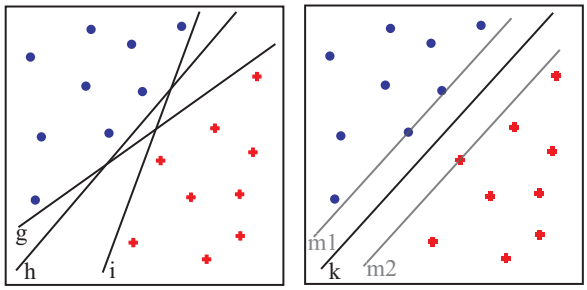
$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left\{y_i\left[(\mathbf{w}\cdot\mathbf{x}) + b\right] - 1\right\}, \tag{22}$$



(a) data set (b) initial codebook vectors

(c) after 25 learning steps (d) after 60 learning steps

Figure 7: The learning process of the LVQ classifier: (a) the original data set with color coded class labels; (b) the circles are the initial codebook vectors for each of the 3 classes; (c) and (d) display the path of the codebook vectors while they move towards the group of training patterns that have the same class.



(a) (b)

Figure 8: Optimal separating hyperplanes (OSH): (a) g, h, i are valid but not optimal. (b) k is the OSH, the distance between k and m1 respectively m2 is equal and maximal.

where α_i are the Lagrange multipliers. Equation (22) may be transformed into problem (23) which is easier to solve.

$$\bar{\mathbf{w}} = \sum_{i=1}^{l} \bar{\alpha}_i y_i \mathbf{x}_i, \qquad \bar{b} = -\frac{1}{2}\bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s] \tag{23}$$

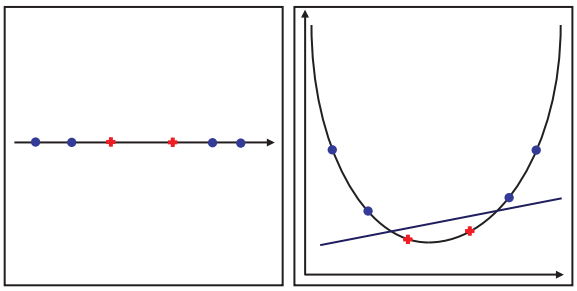
where $\mathbf{x}_r$ and $\mathbf{x}_s$ are two arbitrary support vectors with $\bar{\alpha}_r, \bar{\alpha}_s > 0, y_r = 1, y_s = -1$. Slack variables ζ_i and a penalty function $F(\zeta) = \sum_{i=1}^{l} \zeta_i$ are the means by which SVMs become applicable for the non-separable case [11]. The separating 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 9 illustrates the effect of a polynomial kernel that maps the input space into a feature space of higher order.

Equation (24) describes the SVM classification, where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel used.

$$f(x) = sign\left(\sum_{support\ vectors} \bar{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + \bar{b}\right) \tag{24}$$

There are three typical kernel functions:



(a) not linearly separable (b) linearly separable

Figure 9: The kernel maps the one dimensional input space (a) into a feature space of higher dimensionality, where the inputs become linearly separable (b).

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 certain parameter values.

Kernel functions are not limited to the ones mentioned above. Any symmetric function that satisfies the conditions in 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. Beside 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. First, the author discusses the performance of the individual features. Features that suboptimally discriminate the classes contained in the test set are considered first. The results of the individual features are depicted in more detail in Section 4.1 The results obtained by the combination of multiple features are illustrated in Section 4.2. Finally, the quality of the classifiers employed in the experiments is compared in Section 4.3.

4.1 Individual Features

The first test series concernes with individual features. Each feature is computed from the test and training set described in Section 3.2. The three selected classifiers (K-NN, LVQ, and SVM) are employed to test each feature. The classifiers are trained by the training set to construct a model of the data. The test data serve to validate the model. After classification recall and precision of the classifiers results are determined. Data from the training set are not incorporated in the evaluation.

Some of the features in the experiments do not perform sufficiently well. They are discussed in Section 4.1.1 and 4.1.2. The well performing features are considered afterwards.

4.1.1 Basic Signal Processing Transforms

In the experiments the author employs basic signal processing transforms such as DFT, DCT, and DWT to compute features. The experiments show that the first few transform coefficients of DFT, DCT, DWT, and CQT insufficiently discriminate the animal sounds. The reason for this is that significant information in higher frequency bands is not contained in the first transform coefficients. In the case of animal sounds, high frequencies contain significant information (e.g. for cats and birds).

Table 1 shows mean recall and mean precision over all classes obtained by the classifiers in the experiments. The classifiers represent the rows of the table and the different feature sets are arranged in the columns. A recall

of $\leq$ 0.25 denotes that the corresponding feature does not discriminate the classes. Since the test data contains four classes, a recall 0.25 is equally well to guessing the class labels. In Table 1 such results are marked italic.

$\forall$	DFT		DCT		DWT	
classes	mean R.	mean P.	mean R.	mean P.	mean R.	mean P.
K-NN	0.51	0.51	0.49	0.50	*0.25*	*0.06*
LVQ	*0.25*	*0.05*	*0.25*	*0.06*	*0.25*	*0.06*
SVM	0.48	0.49	0.29	0.37	*0.25*	*0.05*

Table 1: Mean recall and mean precision obtained with the features extracted from the signal processing transforms (DFT, DCT, and DWT). The rows show the results for different classifiers.

The best results for the DFT feature are obtained by the K-NN classifier, with $K = 1$. The SVM employed for the DFT feature uses a polynomial kernel of second order. The DCT feature is classified best by the K-NN classifier (again with $K = 1$). The feature data of the DWT cannot be explained by any of the three classifiers.

To rule out the possibility that the number of features respectively the number of coefficients is to small, the dimension of the feature vectors is increased. For example, the first 200 Fourier coefficients yield a mean recall of 0.44 for all classes and a mean precision of 0.5. Results degrade by further increasing the dimension of the feature vector. The same behavior can be observed for DCT coefficients.

Contrary to expectations, DWT performed worst, followed by the DCT. The DFT feature yields the best results in context of the basic signal processing transforms.

4.1.2 Spectral Flux and Short-Time Energy

In contrast to the high dimensional features discussed in Section 4.1.1, Spectral Flux (SF) and Short-Time Energy (STE) have a dimension of two. Performance of low-dimensional features usually is below that of high-dimensional features, because low-dimensional features are not able to sufficiently represent the samples.

Mean recall and mean precision for all classes of SF is about 0.47. This result is poor at first sight. Compared to other features such as the DCT feature described in 4.1.1, SF yields competitive results, despite its much lower dimension. SF is more meaningful than the basic signal processing transforms but SF alone cannot discriminate the sounds. In combination with other features SF may improve results as illustrated in Section 4.2.

STE is only useful in classification based on frames. When STE is computed for entire files, it represents the average energy of the sound sample, which does not provide meaningful information.

4.1.3 Zero Crossing Rate

The Zero Crossing Rate (ZCR) is a measure for the dominant frequency in a signal. Despite its dimension of one, ZCR provides good discrimination of animal sounds, illustrated in Table 2.

ZCR	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.79	0.77	0	0	0.76	0.80
cat	0.53	0.59	0	0	0.65	0.67
cow	0.37	0.37	1	0.23	0.41	0.48
dog	0.40	0.36	0	0	0.57	0.45

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

Cats and birds are discriminated best by the ZCR. The distinction between dogs and cows causes problems for the ZCR. Analysis of the retrieval results of SVM show that 37% of the cows are recognized as dogs and 33% of the dogs are classified as cows. The reason for the bad separation of dog sounds and cow sounds may be the similar frequency characteristic of these two classes.

The SVM with linear kernel provides the best overall results. LVQ is not able to distinguish between classes at all. It recognizes all animal sounds as sounds of cows. The low dimension and the low computational complexity

of the ZCR, qualify this feature for the distinction of animal sounds in combination with other features.

4.1.4 Pitch

Pitch is a perceptual feature that represents the perceived frequency of a sound. The statistical moments of first and second order are used as features. The results are depicted in Table 3.

Pitch	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.34	0.43	1	0.26	0.06	0.6
cat	0.63	0.71	0	0	0.71	0.78
cow	0.71	0.64	0	0	0.92	0.48
dog	0.68	0.53	0	0	0.68	0.58

Table 3: Results (recall and precision) of Pitch for each class (rows), obtained by different classifiers (columns)

The results are promising for cats, cows, and dogs. The class of birds is poorly discriminated. Against one's expectations, a majority of 63% of birds are classified as cows by the SVM. This results in a low recall for the class of birds. The K-NN classifier yields more balanced results but with a similar distributeion. Again, LVQ is not applicable with this feature.

4.1.5 Constant Q Transform

The constant Q Transform (CQT) is an extension of the FT developed for music information retrieval. Its coefficients are used as features. Retrieval results of the CQT are given in Table 4

The CQT separates birds and dogs well but provides a low precision for these classes. This indicates that multiple samples of other classes are assigned to these classes. Analysis of the retrieval data approves this assumption. A majority of cats and cows are recognized as dogs and birds.

Although, the mean recall of 0.65 over all classes and a mean precision of 0.75 obtained by the K-NN classifier pretend to be fair average, more spe-

CQT	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.93	0.55	0	0	1	0.30
cat	0.38	0.95	0.55	0.40	0.04	1
cow	0.41	1	0.60	0.60	0.04	1
dog	0.86	0.53	0.90	0.53	0.24	0.50

Table 4: Results (recall and precision) of CQT for each class (rows), obtained by different classifiers (columns)

cific analysis of the data shows that the retrieval quality among the classes is not balanced well. The CQT does not qualify for animal sound retrieval.

4.1.6 Sone

The Sone feature contains perceived loudness information of 40 frequency bands. The results obtained by this feature are shown in Table 5.

Sone	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.71	0.74	0.91	0.68	0.80	0.67
cat	0.39	0.60	0.45	0.83	0.37	0.61
cow	0.71	0.59	0	0	0.64	0.83
dog	0.74	0.56	0.97	0.42	0.90	0.58

Table 5: Results (recall and precision) of Sone for each class (rows), obtained by different classifiers (columns)

Sone performs well on the data set. All classes except cats yield relatively high recall and precision. The author remarks that a hit rate between 60% and 70% is a satisfactory retrieval result for environmental sounds respectively animal sounds. The SVM with a linear kernel performs best on the Sone feature. Although it well separates birds, cows, and, dogs, it is weak in recognizing cat sounds. 24% of the samples that contain cat sounds are predicted as birds and 30% are considered as dogs. The results of the K-NN classifier are slightly lower than that of SVM and are more balanced. K-NN

shows similar weaknesses as SVM for this feature. LVQ yields results below that of SVM and K-NN.

4.1.7 Perceptual Linear Prediction

Perceptual Linear Prediction (PLP) is a feature introduced and successfully applied in speech recognition. As Table 6 shows, PLP may also be applied for discrimination of animal sounds.

PLP	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.89	0.81	0.94	0.64	0.69	0.65
cat	0.35	0.87	0.45	0.88	0.36	0.51
cow	0.64	0.58	0	0	0.59	0.65
dog	0.78	0.49	0.99	0.45	0.72	0.51

Table 6: Results (recall and precision) of PLP for each class (rows), obtained by different classifiers (columns)

The K-NN classifier with $K = 5$ performs best on the PLP feature data. It separates the class of birds very well (recall = 0.89 and precision = 0.81). The results for the classes cow and dog are moderate. Sounds of cats pose a problem for all three classifiers in the test. A majority of cat sounds are recognized as dogs. Dogs in turn are confused with cows and vice versa.

In context of the test database of animal sounds, PLP is mainly applicable to the distinction of birdsong from other animal sounds. Besides, it must be pointed out that PLP is a speech recognition technique, that was not designed to distinguish between animal sounds.

4.1.8 RASTA-PLP

Rasta-PLP is an extension of PLP that considers additional properties of human hearing. The RASTA technique improves the results of retrieval, as illustrated in Table 7.

The results of RASTA-PLP are similar to the results of PLP. Again, the class of cat sounds is not discriminated well while most birds are correctly

RASTA-PLP	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.82	0.73	0	0	0.78	0.8
cat	0.48	0.58	0	0	0.47	0.67
cow	0.61	0.72	1	0.24	0.74	0.60
dog	0.81	0.64	0.08	1	0.78	0.65

Table 7: Results (recall and precision) of RASTA-PLP for each class (rows), obtained by different classifiers (columns)

classified. In contrast to PLP, the RASTA technique improves the recognition of cows and dogs. SVM and K-NN perform equally well. LVQ does not produce feasible data. A mean recall of 0.69 and a mean precision of 0.68 are satisfactory values for a single feature in this domain.

Whether the additional computational costs of the RASTA technique justify the improved retrieval quality or not, is dependent of the application domain. With respect to the data set used, RASTA-PLP is more reasonable than PLP.

4.1.9 LPC

Similarly to the PLP technique, LPC is a popular feature in speech recognition. Furthermore, it is utilized for signal compression. LPC coefficients may be represented in many different ways such as autoregressive coefficients, cepstral coefficients, and reflection coefficients [5]. For the data set used, the representation as impulse response is the best choice. 20 LPC coefficients are extracted from each sound sample. The results are illustrated in Table 8.

LPC is the first feature in this experiments that yields average recall and precision above 70%. LPC coefficients discriminate all classes well. Best results are gained by the SVM with an RBF kernel. The average recall and precision over all classes are about 80%. SVM only experiences problems with the dog class, where 21% of the sounds are classified as cat sounds. All other classes are well separated. The NN classifier suboptimally explains the data in comparison with SVM. Especially the cat class cannot be dis-

LPC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.74	0.81	0.94	0.92	0.91	0.92
cat	0.52	0.63	0.86	0.74	0.85	0.73
cow	0.76	0.76	0.73	0.95	0.76	0.88
dog	0.88	0.64	0.74	0.74	0.69	0.74

Table 8: Results (recall and precision) of LPC for each class (rows), obtained by different classifiers (columns)

tinguished well from the other classes with NN. Contrary to the experiences gained in the previous experiments, LVQ demonstrates high performance, comparable to the other classifiers. Especially the results in Table 8 are even better than that of SVM. Since LVQ chooses its initial codebook vectors randomly, the results in Table 8 are not as significant as the results of deterministic classifiers such as K-NN and SVM. In another run, LVQ obtained a mean recall and precision about 70%.

The results of LPC for the distinction of animal sounds are surprising, because LPC is usually employed in speech recognition. The experiments show that LPC may be successfully applied in other domains as well and must not be debared from environmental respectively animal sound recognition.

4.1.10 MFCC and BFCC

MFCC and BFCC are among the most popular features for audio analysis and speech recognition. The first 20 MFCCs and BFCCs are considered as features [5]. Delta and double delta cepstral features perform poorly and are not used. MFCCs and BFCCs perform nearly identically. This results from the fact that both are cepstral domain features that only differ in the psycho-acoustical scaling. Tables 9 and 10 depict the retrieval results of MFCC and BFCC.

MFCCs deliver the best results using the K-NN classifier with $K = 1$ (mean recall of 0.81 and mean precision of 0.83). That indicates that MFCCs cluster the feature space according to the classes. The SVM with a linear

MFCC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.82	0.91	0.79	0.83	0.78	0.84
cat	0.70	0.83	0.79	0.69	0.80	0.68
cow	0.81	0.97	0.77	0.91	0.77	0.98
dog	0.90	0.60	0.72	0.70	0.82	0.75

Table 9: Results (recall and precision) of MFCC for each class (rows), obtained by different classifiers (columns)

kernel yields similar results for MFCCs. LVQ provides slightly lower performance for these features.

MFCCs are well suited to discriminate the classes of animal sounds. Analysis of the wrong classified test samples show that a majority of cat and cow sounds are assigned to the dog class by the NN classifier. The SVM detects 20% of the bird sounds as cats and only 12% of cats as dogs. Most wrong classified cow sounds are assigned to the class of cat sounds by the SVM.

BFCC	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.84	0.87	0.83	0.77	0.81	0.77
cat	0.77	0.86	0.79	0.73	0.74	0.74
cow	0.73	0.89	0.74	0.85	0.73	0.98
dog	0.93	0.67	0.74	0.78	0.90	0.75

Table 10: Results (recall and precision) of BFCC for each class (rows), obtained by different classifiers (columns)

As expected, BFCCs perform equally well in comparison with MFCCs. Again the K-NN classifier ($K = 1$) performs best, followed by SVM and LVQ. Best classification is obtained for the class of dogs. The distribution of the misclassified samples is similar to that of MFCC. Bird sounds are confused with cat sounds. Furthermore 13% of cats are assigned to the dog class. 12% of cows are recognized as cats.

The experiments show that MFCC and BFCC are equally well applicable for the distinction of animal sounds. They can be efficiently computed and a relatively low number of coefficients suffice to yield satisfactory results. An average recall and precision above 80% is obtained with SVM, using only the first seven MFCCs. That indicates that the succeeding coefficients do not contain information useful for classification by the SVM. For the first seven BFCCs results slightly degrade to a mean recall of 0.77 and a mean precision of 0.79. Most information is contained in the first two MFCCs respectively BFCCs. Classification based on the first two coefficients yields results that explain about 70% of the data (mean recall and mean precision about 0.7). These results are very good considering the low dimension of the feature.

The experiments do not show crucial advantages of one of the features. The combination of both features does not increase retrieval quality because they characterize the same properties of the signal. Their computational complexity is equal. The choice between MFCC and BFCC depends on the application domain, they are employed in.

4.1.11 Amplitude Descriptor

The last feature tested is the Amplitude Descriptor (AD) introduced in [38]. The AD consists of LoHAS (mean, standard deviation, median), LoLAS (mean, standard deviation, median), and AHA (mean). Results obtained by the AD are given in Table 11.

AD	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.80	0.79	0.84	0.65	0.90	0.76
cat	0.54	0.78	0.72	0.70	0.68	0.81
cow	0.74	0.78	0.49	0.93	0.72	0.78
dog	0.93	0.64	0.78	0.70	0.86	0.81

Table 11: Results (recall and precision) of AD (LoHAS, LoLAS, and AHA) for each class (rows), obtained by different classifiers (columns)

The simple but intuitive features contained in the AD, capture the properties of the specific classes well. The combination of LoHAS, LoLAS and AHA is able to explain most of the test data. Classification with the SVM and a linear kernel provides an average recall and precision of 0.79. All classes are separated well from each other. The minimum recall is 0.68 for cats. Dogs are separated best from the other classes. Only 8% of dogs are assigned to cows and 5% are classified as cats. Most misclassified cows (11%) are recognized as cats. Cats are suboptimally separated. 19% of all cat sounds are predicted to be bird sounds by the SVM. 90% of all birds are correctly classified. The remaining 10% of bird sounds are uniformly distributed over the other classes. The precision is high for all classes (above 0.76).

Satisfactory results are also achieved with LVQ and K-NN. The results of AD are comparable to other well performing features such as MFCC, BFCC and LPC. For the NN classifier recall and precision of AD lie between those of LPC and MFCC. The results of LVQ for the AD are slightly below that of the other well performing features.

4.2 Combined Features

Up to now the author concentrated on individual features. In order to improve retrieval quality, several features are combined to a feature vector. This makes sense because the combination aggregates information present in separate features. In the second test series, the author experiments with all features employed in the first series again. The possibility cannot be eliminated that a weak feature attains synergy together with another feature. That is why even poorly performing features such as Zero Crossing Rate and Spectral Flux are considered.

An optimal solution to the retrieval problem is empirically determined. Therefore the author first combines the best performing features and observes the synergy effects between them. Features that do not improve retrieval quality are not selected. Then weaker features are added to the selection. Again, only features that improve results remain in the combination. An SVM is employed to evaluate the discriminative power of the combinations. The SVM is chosen because it is well applicable to high di-

mensional feature vectors. Such feature vectors may emerge in this series of the experiments when multiple features are combined to a high dimensional vector.

This strategy finally yields a feature vector that comprises 26 components. The first six components are mean, standard deviation, and median of LoHAS respectively LoLAS. The first LPC coefficient is left out because it always contains a constant value. The succeeding four LPC coefficients are added to the selection. Additionally the first 13 MFCCs are selected. Further components are the mean SF, the mean Pitch, the first RASTA-PLP coefficient and the mean of Sone. The results of the feature combination are illustrated in Table 12.

Combination	K-NN		LVQ		SVM	
	Recall	Precision	Recall	Precision	Recall	Precision
bird	0.86	0.94	0.98	0.91	0.97	0.94
cat	0.83	0.81	0.83	0.79	0.88	0.91
cow	0.82	0.96	0.72	0.88	0.81	0.97
dog	0.90	0.74	0.78	0.75	0.96	0.80

Table 12: Results (recall and precision) of the combined feature vector for each class (rows), obtained by different classifiers (columns)

Classification based on this feature vector yields an average precision and recall above 0.9 using the SVM with a linear kernel. This is a significant improvement over results with individual features. There are only a few misclassifications worth mentioning. 8% of cats are assigned to the class of dogs. 9% of cow sounds are predicted as dog sounds and 8% as cat sounds. All other classes are well separated and misclassifications are below 3%. LVQ and K-NN profit from the combined feature vector as well. LVQ correctly predicts more than 80% of the test samples' classes. K-NN obtains an average recall of 0.85 and an average precision of 0.86.

The combination of features increases retrieval quality. At the same time computational complexity increases because more features have to be computed. Especially in mobile applications there is always a tradeoff between retrieval quality and computational costs.

4.3 Comparison of Classifiers

All classifiers in the experiments achieve satisfactory recall and precision (between 0.7 and 0.9). The classifiers do not perform equally well on different features. For very poor features such as DFT and DCT, only SVM and NN yield consistent results. LVQ is not able to explain any of the data obtained by poor features. Useful results with LVQ are mainly achieved for well performing features such as MFCC and LPC.

SVM is different from the other classifiers used. K-NN and LVQ depend directly 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 more generalization ability, while with a model of high order, classification runs the risk of overfitting.

SVM and K-NN perform comparably on the test data. We cannot identify a winner among the two classifiers. For high dimensional feature vectors SVM usually outperforms K-NN. The feature vectors in the experiments have relatively low dimension. This may be the reason for the similar performance of SVM and K-NN in the investigations.

In most experiments K-NN is used with $K = 1$. K-NN is tested with different $K > 1$ to find 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 animal sounds. The linear kernel is well suited for most features. That indicates a clear structuring of the feature data. The RBF kernel demonstrates good performance on the LPC feature. Polynomial kernels are outperformed by linear and RBF kernels.

Computational complexity of the classifiers investigated are different. Measurements refer to the time the classifier takes for training and classifying the test samples. Classification of the combined feature using the SVM with a linear kernel takes 63 milliseconds. The K-NN classifier takes 78 milliseconds for the same task. LVQ is much slower at classifying. Useful results

are obtained from a minimum of 15 epochs. The LVQ takes 1.38 seconds for 15 epochs, where most of the time is spent on training. All three classifiers are well suited for frame-based classification. Furthermore, they may be employed in mobile respectively realtime applications. In contrast to LVQ, SVM and K-NN are able to be (re)trained very quickly.

5 Related Work

There are different groups 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 [54]. 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). Classification is accomplished by the nearest neighbor rule.

An alternative to the direct measure of similarity is the use of artificial intelligence techniques such as Support Vector Machines (SVM) [11], Hidden Markov Models (HMM) or Artificial Neural Networks (ANN). An early example in the domain of audio processing is presented in [19]. 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 [45] separate music and speech with low level features. They apply Spectral Centroid, Spectral Flux, Zero Crossing Rate, 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 are used to separate speech from music based on the features. 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 [58] 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 [41] researchers focus on recognition of the spoken language [39]. Another field of research is classification of the speaker (e.g. for customization issues or authentication) [43]. In the area of multimodal dialog systems, recognition of human emotions from audio gained focus [8]. The different areas of speech processing are a source to survey state-of-the-art audio features.

Beside speech recognition, music information retrieval (MIR) 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. Multiple speech recognition features are applicable to the classification of music. In [33] the authors distinguish between instruments (e.g. Brass, Keyboard, and String) by extracting features such as ZCR, STE, Bandwidth, Pitch, Formant Frequencies and MFCCs. 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 final feature vector that represents the signal. Classification is performed by GMM and NN. Instrument recognition is proposed in [37]. 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 [24]. 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 [36]. 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 (OSCCs). An SVM trained with OSCC features separates vocal from instrumental sections.

Environmental sound recognition concerns with the analysis 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 [34] 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 an appropriate neural network for each class. A well investigated problem is highlight detection in sport videos. The authors of [47] 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 [55]. 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 [56] 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 [57] multilevel classification is proposed. First the authors apply a coarse level segmentation to separate speech, music and environmental sound. In a second step an HMM is considered to analyze environmental sounds (e.g. footstep, laughter, rain, windstorm). The authors of [30] present an audio indexing system using MPEG-7 features. 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 [22]. The authors compare and combine cepstral features (MFCCs) with perceptual features (Total Spectrum Power, Subband Powers, Brightness, Bandwidth, and Pitch). In [22] perceptual features outperform cepstral features. Best results are obtained by a combination of both. In [22] 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 around a human user. From this information a diary is built 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 [32]. "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 used 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. Applications discussed in this section prove the importance of environmental sound recognition for future information systems.

6 Conclusion & Future Work

Discrimination of animal sounds is a rarely considered area of environmental sound recognition. While some investigations on environmental sound recognition involve animal sounds among other sounds, there is few work on the discrimination of animal sounds from each other. This thesis concerned with content-based retrieval of animal sounds. Due to the lack of a publicly available database of animal sounds, the author collected numerous audio samples from the internet and built a database that contains about 380 animal sounds arranged in four classes. A survey of widely used audio features and classifiers was presented. The research focus was the investigation of their applicability for animal sound recognition. The experiments show that popular features employed in speech recognition such as LPC coefficients and MFCCs separate the classes of animal sounds well. Furthermore, the author observes that low complex features such as Fourier coefficients and Wavelet coefficients poorly perform on animal sounds.

The author introduced a set of novel time-based audio features. They follow an intuitive way to describe the characteristic shape of a waveform. Despite their simplicity, they perform comparably to much more complex features, such as MFCC and LPC. The investigation shows that a combination of state-of-the-art features with the feature set introduced by the author, enables to successfully classify more than 90% of the animal sounds contained in the database.

The author employed three popular classifiers in the experiments. The SVM and the K-NN classifier perform equally well. Both achieve high precision and recall and even consistent results for poorly performing features. 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 on weak features are poor.

The results of the investigation are promising for future research in this area. Frame-based classification may further improve results of file-based classification. Therefore, context sensitive classifiers such as Hidden Markov Models and Artificial Neural Networks will be employed. Further work will

include the comparison of features discussed in this thesis with MPEG-7 features in the domain of environmental and animal sounds.

Another future goal is the distinction of different sounds from the same species ("understanding animals"). Such a technique may be useful in analyzing animal behavior. It may also improve the understanding of humans for their animals. Besides, a focus will be the design of new audio features for environmental sounds. An interesting area are mobile information systems such as life logging and supportive systems for handicapped people.

Appendix

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

A Implementation

This section contains Java implementations of features employed in the investigations. In Section A.1 the Java source code of the Amplitude Descriptor introduced in [38] is presented. In Sections A.2 and A.3 the author provides source code for Short-Time Energy and Zero Crossing Rate.

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 treshold
    float treshold =
       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] >= treshold && new_LoHAS) {
            counter_HAS = counter_HAS + 1; //continue HAS
            accumulator_AHA =
             accumulator_AHA + (this.mSignal[i]-treshold); //increase AHA

        }
        else if (this.mSignal[i] >= treshold && !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]-treshold;
```

```
        }
        else if (this.mSignal[i] < treshold && new_LoLAS)  {
            //    continue with LAS
            counter_LAS = counter_LAS+1;
        }
        else if (this.mSignal[i] < treshold && !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

```
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*, pages 10–15, January 2005.

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

[3] C. Borgelt. Learning vector quantization visualization. *http://fuzzy.cs.uni-magdeburg.de/∼borgelt/doc/lvqd/#Download*, 2000.

[4] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *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*, pages 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, March 1999.

[8] C. Chiu, Y. Chang, and Y. Lai. Analysis and recognition of human vocal emotions. *In Proceedings of the International Computer Symposium*, December 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*, pages 130–138, 2005.

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

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):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(5):961–1005, 1990.

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

[16] H. Eidenberger. A new perspective on visual information retrieval. *SPIE Electronic Imaging Symposium*, 5304, 2004.

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

[18] 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(17-21):665–668, May 2004.

[19] B. Feiten and S. Gunzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18(3):53–65, Summer 1994.

[20] 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.

[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, August 1997.

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

[23] 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.

[24] M. Grimaldi, Cunningham P., 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.

[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(4):578–589, October 1994.

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

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

[30] 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*, pages 716–725, 2004.

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

[32] 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*, February 1994.

[33] 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.

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

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

[36] 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.

[37] 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)*, October 1998.

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

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

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

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

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

[43] 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, January 1995.

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

[45] 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)*, pages 1331–1334, 1997.

[46] 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)*, May 2002.

[47] 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.

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

[49] 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.

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

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

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

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

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

[55] 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.

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

[57] 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, March 1999.

[58] 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, May 2001.