FP6-NEST-PATH project no: 29085
Report Version: 1
Report Preparation Date: July 07, 2008
Classication: Pub.
Deliverable no. 5.1

Closing the Loop of Sound Evaluation and Design (CLOSED)

Deliverable 5.1

Representations and Predictors for Everyday Sounds

Kamil Adiloğlu, Robert Anniés,
Hendrik Purwins, Klaus Obermayer
Neural Information Processing Group (NIPG)
School IV – Electrical engineering and Computer Science
Technische Unversität Berlin

Contents

Notation

fraktur script ($\mathfrak{s}, \mathfrak{M}, \ldots$) denote measures, quantities or functions in the domain of psychology, cognition or psychophysics. It denotes something that has to be modeled in order to make it mathematically tractable. Thus, there is only a qualitative definition.

bold letters ($\mathbf{x}_i$) denote vectors in $\mathbb{R}^n$

italic letters (x_i) scalar values

capital letters (M) sets, $G = (V, E)$ denotes graphs, where V denotes the vertices and E denotes the edges.

bold capitals ($\mathbf{T}$) matrices

indices ($_i, ^\alpha$) $i, j, \ldots$ index examples and the upper index α different contexts or models, which is omitted where appropriate, y^T denotes a target value, which is the known label of an example $\mathbf{x}$.

angle brackets inner/dot/scalar product: $\langle \mathbf{x}, \mathbf{y} \rangle$

1 Introduction

This report documents the work of NIPG within the CLOSED project according to the tasks T5.1 and T5.2 defined in Workpackage 5: *Measurement Definition.* Within CLOSED we have investigated what methods of machine learning, signal processing and statistics are applicable to construct predictors for everyday sounds. These predictors can be used in computer assisted sound design and in research on human perception and recognition. This includes to find a suitable representation for everyday sounds.

The context of this research is the work of our partners, who provided their results as input to us. In particular we used the developed sound synthesis models in D2.1 [34] by UNIVERONA, D3.1 [51] on scenarios and interaction by ZHdK and the teoretical work in D4.1 [19] by IRCAM. This work was essential to us to work out an adequate framework and carry out the experiments on everyday sound classification.

Perception and classification of everyday sounds is a task, which human beings perform each day. In D4.1 was shown that the context and experience and the ability to categorize play an important role in perception and classification of everyday sounds. In D2.1 an everyday sound taxonomy was proposed from a generating point of view. Depending on this taxonomy UNIVERONA delivered the sound design tools in terms of physically-based sound synthesis algorithms. D3.1 influenced our work particularly in the research of online sound adaptation of an artifact that incorporates human interaction.

In this deliverable, we present the psycho-acoustically relevant features in the classification of the everyday sounds – with respect to the above sound taxonomy – and the methods we used to extract these features. Binary and multi-class classification experiments have been performed using different machine learning algorithms, in order to test the performance of the representation methods.

The classification experiments can be performed by making use of a sound database prepared beforehand. However, in a psychoacoustical experiment, the sounds are presented to the subjects online. In these kinds of experiments, subjects listen to the sounds randomly, and evaluate them in an online fashion. Therefore, we have been working with IRCAM together on a new online experimental paradigm, which selects the sounds to be presented to the subjects in a more clever way than random selection. The so-called active learning scenario has been adapted to perform psychoacoustical experiments.

Online learning proposes many possibilities not only in a classification scenario. Online classifiers and predictors, based on psychoacoustically relevant features, which reduce the number of data points (sounds) to perform an experiment enables us to develop optimization algorithms for the sound design tools developed by UNIVERONA in WP 2. These optimization algorithms are utilized in interactive scenarios and sound products developed by ZHdK in WP 3 used in psychoacoustical sound optimization experiments. In D4.2, IRCAM develops experimental methods to study the suitability of a sound product for a function in these interaction scenarios. Consequently, online optimization algorithms can be incorporated to combine all these research studies to improve the quality of the sound design tools considering their support of the sound products in an interactive experimental scenario.

In this report, the general frame work related to the theoretical work on perception is presented in Chapter 2. Chapter 3 documents the work on sound classifcation and representation. A practical case study incorporating the sound design tools in an online experimental paradigm is introduced in Chapter 4. Chapter 5 explains the details of the online interactive optimization in another case study.

2 General framework

Perception of everyday sounds is an upcoming topic, which is in the focus of psychoacoustical research as well as music and speech perception. Firstly to learn more about sound perception and recognition in natural and urban environments, secondly to gain knowledge of how to adapt sound generating processes of technical devices to support their usability, design sounds for auditory displays or reflect or induce certain emotions of the listener. To achieve this we need to analyze the relation between sound and human recognition and perception. Training predictors is a way to study this relationship and to develop measurement tools of perceptual parameters.

In a top-down approach the connection between an acoustic signal and perceptual categories of sound events, sources or emotional responses is modeled. We will formalize the ideas mathematically as far as possible to make it tractable for computational methods, so that we are able to propose methods for sound design and analysis.

2.1 Perception and Recognition of Sound

The human perception of the world depends strongly on the ability to categorize. In psychophysics and cognition this research is generally summarized under the term *categorical perception.* Research on grouping of colors [10], [3] and categorical perception of speech phonemes [26] show the principle, that the perception discretizes continuous stimuli, such that the discriminability within categories is small and between categories is large. Boundary effects appear [35], where perception jumps from one to another category without effecting a perception of a continuous transition (e.g. between the phonemes /ka/ and /pa/).

Categorical perception theory addresses questions about the origin of categories. Are they innate or learned and if yes, how are they learned? Are they induced by physical/physiological facts or by cognitive decisions? Can boundaries be adapted by training? Why does categorical perception exist at all?

A thorough literature review was given in D4.1 of WP4 [19] section 1.2, which emphasizes the importance of categories for human perception and recognition. It was discussed and concluded, – and this is the basis of D5.1 – that similarity and categorization are the basic building blocks of perception of everyday sounds.

Let's consider first sound itself: physically the stimulus can be described completely as temporal signal

$$s(t)\,; \quad t = [0, T],$$

let it be finite. Perception of a sound is the transformation of the signal arriving at the perceiver (the ear) into another form that can physiologically better assimilated.

$$\mathfrak{p} : s(t) \mapsto \mathfrak{s}, \quad \mathfrak{s} \in \mathfrak{S},$$

where $\mathfrak{S}$ is the set of all stimuli. The result of this transformation is not an oscillation in the Maxwell sense anymore, but a multi channel neural spiking code or just a pattern of excitations of a neural network. Recognition is a match of an excitation pattern in the memory that triggers envisioning a situation, object, action or leads to an emotional response. As the studies on *categorical perception* show, we can consider those effects as a countable and finite sets of phenomena:

$$\mathfrak{M} = \{\mathfrak{C}_{p_1}, \mathfrak{C}_{p_2}, ..., \mathfrak{C}_{p_N}\},$$

where $\mathfrak{C}_{p_i}$ denotes a perceptual category indexed by p_i: a descriptive label for the category, like *car*, *train*, *bike*, in a vehicle recognition task. A recognition in this setting is a mapping of stimuli to a predefined set categories:

$$\mathfrak{h} : \mathfrak{S} \to \mathfrak{M}$$

Since categories can be hierarchical and recognition tasks are context dependent, it is necessary to restrict the possible categories.

- discrimination of sound sources: $\mathfrak{M}^1 = \{\mathfrak{C}_{\text{car}}, \mathfrak{C}_{\text{train}}, \mathfrak{C}_{\text{bike}}\}$
- induction of emotional responses: $\mathfrak{M}^2 = \{\mathfrak{C}_{\text{annoying}}, \mathfrak{C}_{\text{pleasant}}, \mathfrak{C}_{\text{boring}}\}$
- recognition of actions and functions: $\mathfrak{M}^3 = \{\mathfrak{C}_{\text{opening}}, \mathfrak{C}_{\text{closing}}\}$
- detection of occurrences: $\mathfrak{M}^4 = \{\mathfrak{C}_{\text{fire alarm}}, \mathfrak{C}_{\text{no fire alarm}}\}$

Recognition is not only the mapping from stimuli to category, but both the selection of a set $\mathfrak{M}_i$ which restricts the possible answers and a $\mathfrak{h}_i$ process that is specialized according to the context and/or task:

$$\mathfrak{h}^\alpha : \mathfrak{S} \to \mathfrak{M}^\alpha$$

The selection of α is dependent on the environment, attention, experience or expectation. In case the selection is wrong we can expect misclassifications, irritations or the inability to recognize sounds that are known in other contexts: the sound of a motor bike inside an air plane could not only be very irritating, it would take significantly longer to recognize it as such, or even rather lead to a misclassification, e.g. as an air plane engine malfunction. Switching the context by using the ear plugs to watch the on-board film of the same air plane leads to a complete change of expectations and the recognition ability of motor bikes.

2.2 Predictors as Human Perception/Recognition Model

2.2.1 Prediction

Machine learning research addresses the question how automatic systems can learn to classify measurable objects (e.g. objects causing acoustic stimuli) by analyzing examples of these objects using so-called *features* in numerical form that can be measured.

This gives us the ability to model learning and recognition and to investigate them quantitatively. Machine learning algorithms are often motivated by biological and psychological findings. Therefore, for these cases, we can interpret the results provided by a machine learning system back to those domains.

Contrary to the perceptional viewpoint, where learning, predisposition of categories and their boundaries are known to exist, but their location, nature and origin remains an unknown or at least a speculative aspect of perception, in machine learning these are openly examinable since here the entities *class* (category) *discrimination border* (boundary) and learning are the basic building blocks. It is not in question here whether there are classes and boundaries, but how we can model and determine them. Assuming recognition and perception is closely related to perceptual categorization, machine learning gives us the right tools.

We want to focus on supervised learning and classification tasks, where a known set of labeled examples are used to find a generalized rule to predict unlabeled examples. Let

$$(\mathbf{x}_i, y_i^T) \in \mathbb{R}^n \times M; i = 1, \ldots, N$$

be the examples to learn from and

$$h_w^\alpha : \mathbb{R}^n \to M^\alpha$$

a parameterized classification model. h_w can be directly associated with a recognition function $\mathfrak{h}_{\mathfrak{i}}$, the vectors $\mathbf{x}_i$ are modeling the neural activity pattern $\mathfrak{s}$ and y_i is the category number in $\mathfrak{M}^\alpha$, where the matrix M^α contains these numbers.

The model is restricted to a distinct context or task $(\mathfrak{M}^\alpha, \mathfrak{h}^\alpha)$. The examples are chosen to be originating from that context only, we will not model here the selection process of the particular context α, but view it as given.

2.2.2 Representation

The perception mapping $\mathfrak{p}$ (section 2.1) translates signals to *brain-compatible* spike patterns. In machine learning we need a translation to a *predictor-compatible* representation:

$$p : s(t) \mapsto \mathbf{r}; \quad r \in \mathcal{X}$$

where $\mathcal{X}$ is called a feature space, which is usually a vector space ($\mathbb{R}^n$) in which we can draw borders and there exist learning algorithms to construct them from examples.

With p a dimensionality reduction is carried out. The high-dimensional signal has to be translated into a low dimensional description, because signals are (1) noisy and contain redundancies, which carry no information with large bandwidth and (2) the intrinsic (information carrying) dimension of the signal is often much lower than that resulting from digital recording techniques[1]. Machine learning suffers from the *curse of dimensionality* — an exponential increase of computational costs with the number of dimensions, which is rather a natural effect than a limitation of the computational methods. Categorical perception perhaps exists, because the brain has a similar problem with continuous high dimensional input: In [46] the hypothesis is proposed that such an information reduction can be viewed as cognitive economy contributing to a high sensimotoric performance. Similarly, efficient coding techniques of audio signals can be incorporated to represent and compress natural sounds, see [45].

The general objective is to find a function p that filters out the classification irrelevant part of the available information in order to construct an efficient predictor. The translation result $\mathbf{r}$ models the physical/physiological stimuli activity pattern $\mathfrak{s}$ and plays it's role in the modeled prediction process.

There exist various techniques for p to encode acoustic signals (not systematic):

Fourier transformation based The Fourier series can describe any signal as additive sines and cosines. Many methods for signal representation start with a discrete Fourier analysis and use the found coefficients for subsequent processing steps. DFT techniques have the advantage to model the spectrum of a signal, which contains information about the audible frequencies and its distribution. The main disadvantage is that there exists a trade-off between temporal resolution and covered frequency bandwidth.Only one can be optimized at a time. This leads to a block-wise analysis of a signal.

low level signal/spectral properties Several properties of signals that can be easily calculated, mostly based on FFT are: Spectral spread, spectral centroid, zero-crossing, frequency roll-off. The interpretation of such values according to the perception of sound is possible, but no unproblematic. E.g. *roughness* is somtimes describe using zero-crossings, since harmonic content tends to have lower a rate than noisy content.

psychoacoustic descriptors Descriptors with a clear interpretation in terms of their perceptive qualities: e.g. *roughness, loudness, timbre* are called psychoacoustic descriptors. The are evaluated empirically by psychoacoustic experiments and are built mostly based on low-level spectral features and other post-Fourier analyses.

physiological/neural codes In order to get closer to the actual hearing process, models were developed using biological/physiological facts of especially the inner ear, but also outer as well as middle ear. By pursuing these approaches, we can assume that the hearing apparatus that had evolved to its current state is specifically adapted to the task of perception and therefore carries out necessary transformations and filterings as a prerequisite to the subsequent cortical recognition processes. Hence,

[1] extreme example: a sine tone has a dimensionality of 3: frequency, amplitude and phase, whereas a recording of one second of a sine tone in CD quality has 44100 dimensions

modeling this can lead also to a efficient way for automatic classification. Here we also have to assume that the prediction model is able to use this information efficiently and that the model captures the relevant information at all.

generative descriptors Descriptors of sounds can also be understood as controlling parameters for sound generating models. E.g. in FFT the coefficients can be used to resynthesize the signal. Sound generating models do not necessarily need to sum up sines and cosines, but can use other means of sound synthesis: granular synthesis, subtractive synthesis (filters), physical modeling. ¡¡¡¡¡¡¡ .mine

structured/graphical descriptors Instead of describing a given sound by using some preliminary descriptors, Cabe and Pittenger [8] or Warren and Verbrugge [52] tried to explain the invariant structures within the sound. Smith and Lewicki [45] defined a shift invariant and efficient representation for describing the frequency and temporal structures within the given sound. In general, complex sounds have a specific temporal structure. Therefore they are recognized as a compound of events, that generate the sound [42]. These events have a certain temporal order. In a structured and / or graphical representation, these events can be induced by their relative distances to each other within the representation. This way higher level structures can be captured.

2.2.3 Classification

Central in the research of perceptual categorization is the question on how categories are formed. "Categorization", as Harnard [17] puts it,"is intimately tied to learning". Findings in from experimental psychology show that there are categories which are innate [] as well as acquired ones [16], [24]. First focusing on the latter: How does one acquire the ability to categorize something? What is learning? Learning can be described by the problem of induction: How can a general and reliable rule be formed, given a finite set of examples?

Machine learning addresses exactly this question. The basic idea is to quantify generality as the *generalization error* and find ways to minimize it. The problem of induction is substituted by an optimization task, which is a well known topic.

The generalization error measures how wrongly a learned rule will behave in the future. Of course, the future is not known, so it has to be estimated, and in turn the generalization error will always be an approximation. Still, the optimization of it will lead to reasonable rules, given these estimates are not completely wrong.

In categorization or classification the generalization error is defined as the ratio of the number of all (by the learned rule) correctly classified examples and all possible examples that exist:

$$E_G = \frac{N_{\text{wrong}}}{N_{\text{all}}} \approx \frac{N_{\text{wrong on testset}}}{N_{\text{testset}}}$$

A good estimate of that is to take a set, which is large enough to test the rule on it. This set must not have been used to learn the rule: only then it is an estimate of generalization error. What happens while learning is to find a model that is consistent with the examples that have been seen and to assume that the future will be the same or at least similar.

Important concepts in the theory of perceptual categories are category boundaries and similarity []. Boundaries divide the space of stimuli into categories, whereas similarity measures distances between stimuli or between stimuli and the boundaries. Both can be modeled using supervised learning of *Kernel machines*, because they optimize directly discrimination boundaries and they use *Kernel functions* which incorporate a similarity measure. These means are a well studied in the machine learning community [] and give us a useful set of tools to investigate category learning of everyday sounds.

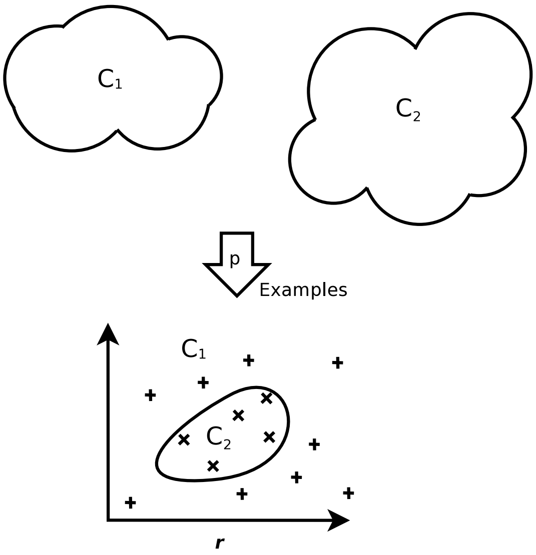

Figure 2.1: The clouds depict two categories of everyday sounds. Drawing examples and transforming them into a feature vector representation makes it possible to learn a decision boundary that discriminates the classes C_1 and C_2 (bottom)

Figure 2.1 illustrates the division of the stimuli space and feature space respectively. The stimuli space is perceptual and not mathematically well defined. Machine learning works in the feature space instead looking for an optimal decision boundary which is expressed using a kernel function

$$\langle \mathbf{w}, \varphi(\mathbf{x}) \rangle + b = 0, \text{where} \quad K(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle$$

where $\mathbf{w}$ and b are adaptable parameters (weights) and $\mathbf{x}$ an example data point in the feature space. The class decision is whether the result is positive or negative. K defines what similarity measure is applied. To interpret K as a similarity measure it has to satisfy the conditions to be (1) a positive semi-definite continuous function and (2) to be symmetric (Mercers theorem: [33]). Dot products (or scalar products) are in fact geometrical similarities as their magnitude depends on the angle between two vectors.

The simplest case is to take $\varphi(\mathbf{x}) = \mathbf{x}$ for which K becomes the standard inner product in the Euclidean space:

$$K(\mathbf{x}, \mathbf{x}') = \sum_i x_i x'_i$$

and the class boundary is simply a linear function: $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$. However, this is a special case. Generally, as depicted, the decision boundary is non-linear using an appropriate Kernel. The role of φ is to transfer a feature vector into a (Hilbert-) space induced by the Kernel. In this space all boundaries are linear and linear optimization and learning algorithms can be applied, which is a great practical advantage. The definition of φ may be hidden, since it is never computed directly. Even the feature space itself can be hidden: If the results of dot product between all pairs of examples is known, the classifier can work with these similarity values alone, since $\mathbf{w}$ depends only on them after learning.

As outlined in section 2.1 we use a context aware approach. As there doesn't exist a global classifier for recognizing all kinds of everyday sounds, because categories are changing with context, task, experience or attention of the subject, we model and train task specific classifiers. For each, we train a classifier, given a finite set of target classes and examples with known perceptual labels (figure 2.2).

The decision which classifier is used must be made by the sound designer, when applied as a measuring tool in sound design – or by a higher cognitive function in terms of recognition. It is important to stress the fact that (enough) examples must have been acquired to train a classifier on a specific task.

Generalization into other domains will not be possible without a proper set of examples, which is especially true for learning in general.

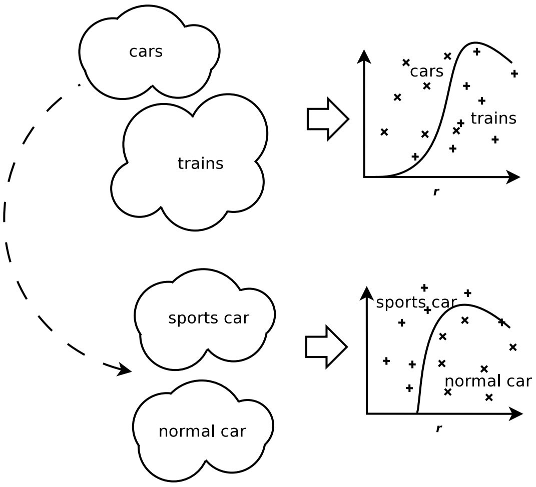

Figure 2.2: The examples form the *cars* class will have changed labels when the context changes (bottom). The two resulting classifiers can use different feature sets $\mathbf{r}^1$ and $\mathbf{r}^2$, as well as as different kernel functions.

2.2.4 Learning

Different approaches exist to find the decision border. Either the border itself is described as a parameterized function and the objective is to find the optimal parameters $\mathbf{w}$ or indirectly by applying a decision rule based on some statistics of the example features.

In this study we use Support Vector Machines (SVM) [5] [50] and Hidden Markov Models (HMM) [2] as batch learning methods and the perceptron algorithm [41] for online learning. The two first methods trains a classifier using all available examples at once, whereas the last is trained using the examples one after another, updating the classifier continuously. We adapted and improved the perceptron method for usage in psychoacoustic experiments using an *active learning* strategy (see Chapter 4 and Section 2.2.4).

The objective is to minimize the generalization error E_G. All information that is available for learning are the examples from the training set, therefore only the Training error E_T can be minimized directly.

We describe shortly the idea of the algorithms focusing on their application to sound classification and design tools.

Perceptron Learning

The classic learning algorithm *perceptron* by Rosenblatt [41] uses a model of single neuron that calculates a weighted some of its inputs and evaluates an activation function with this sum, which is the output value of the perceptron:

$$y = g\left(\sum_i w_i x_i\right) = g(\langle \mathbf{w}, \mathbf{x} \rangle).$$

When using the signum function as activation g this is equivalent to a linear decision boundary in feature space. The perceptron learning rule is defined as:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \langle \mathbf{x}_n, \mathbf{y}_n^T \rangle,$$

which can be extended using a Kernel function for non-linear decision boundaries ([13]) and applying a specific similarity measure other than the Euclidean distance. The training is done by applying the learning rule subsequently using examples that are classified wrongly. The rule adapts the decision boundary at each step controlled by the learning step parameter η.

The convergence was shown for the perceptron and kernel perceptron in case of separable data. It will not converge in the non-separable case. However, in that case learning can be stopped when the test error is not decreasing anymore (early stopping).

Active Learning

For the task at hand subjects will be asked to label sound examples. This is a limiting factor for the amount of labels that can be acquired. Subjects cannot sit for very long time and can label a maximum 300-400 sounds in one session. By using a technique called *active learning* we want to improve the training success with less labeled examples.

The basic idea is to select the examples for labeling in a specific order by an algorithm that estimates the *informativeness* of each of the available unlabeled examples. The algorithm picks only those that is supposedly interesting to know instead to pick examples randomly. A measure on how informative an unlabeled example is was derived as a geometrical problem in [18].

Support Vector Machines

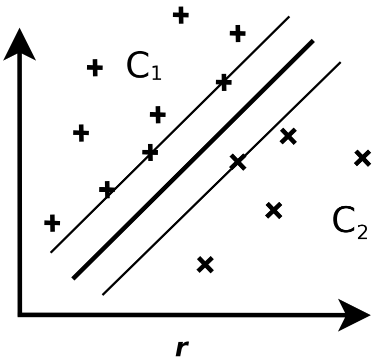

Figure 2.3: Support Vector Machine

As a result of the theoretical considerations about structural risk minimization [50] the support vector machine was developed. It uses the same linear decision border as the perceptron. It finds the parameters of this border by maximizing the *margin*, i.e. the distance between the decision plane and the closest example points and the misclassification rate at once. This optimization has always one unique solution which is more reliable than stochastic processes, like the perceptron.

The decision border depends only – and is constructed from – a few support vectors, which are part of the training set. A prediction is therefore very fast to compute. Like all linear methods the kernel trick is applicable to extend the SVM to non-linear problems or to adopt a specific similarity measure [5].

The SVM is a batch leaning algorithm: The training set is not extendible online like in a perceptron, but incorporated at once in the optimization.

Hidden Markov Models

HMM is a generative approach, which was used mostly in speech recognition. It learns the statistics of a piecewise stationary process from examples that are given as a (time-)series of feature vectors. HMMs models 2 stochastic processes: (1) a Markov decision process and (2) a probability distribution for each state that generates the features.

MFCCs are used usually as features in speech analysis [27]. Figure 2.4 shows a HMM that commonly used. It is a chain of states which are connected by possible transitions.

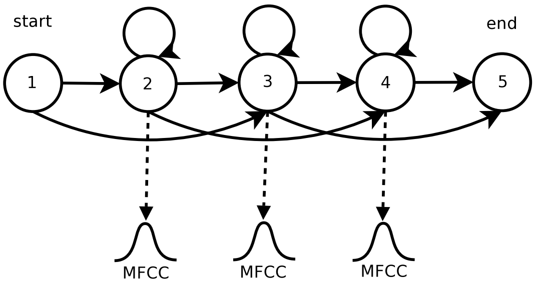

Figure 2.4: Hidden Markov Model

The form of a chain expresses that we want to model a process that develops forward in time. Each state represents a stationary segment of the sound signal. The local loops make it possible to remain in one state, such that the segments are stretchable in time.

The training is done via the Baum-Welch algorithm. Given a set of examples, for each class one HMM is trained. Two things have to be estimated while training: the probability matrix for the state transition and the MFCC distributions of the states, where Gaussian Mixture Models are used. After training the HMMs can be used to classify new sounds. The likelihood is computed for each HMM having produced the given test sound (calculated over all possible transitions). The class label of the model with largest likelihood will be assigned.

A decision border is not trained directly. It is non-linear depending on the used distributions and transitions, but can't be expressed in a closed form.

The advantage of HMMs is that it models a series of features, which is more natural for signals, than using a vector description like in SVMs or perceptrons. Segmentation of sounds are a feature of everyday sounds in particular and can be learned with HMMs directly. However, it comes with the drawback that many parameters have to estimated to model such time series: Not only the transition probabilites have to be estimated but also the (multivariate) distributions in each state. Many examples are necessary to yield good estimators.

3 Classification of Sound

3.1 Representations

With reference to the notation introduced in chapter 2 we seek a representation $\mathbf{r}$ for the acoustic stimulus $\mathfrak{s}$ that contains all information that we need to classify sounds according to the context α of a given classification task.

The representation should take following problems into account:

1. high dimensionality and duration differences of sampled audio signals

   When using digital sound recordings the dimensionality of the data at hand depends rather on the used sampling rate and recording duration than on the actual content. We need features that are independent from that. It is not sufficient to restrict the sampling rate and recording duration to standard values, since then important information can get lost: when using low sampling rates high frequency content is lost (Nyquist), using a fixed duration raises the question which one, and we are not able to provide a general solution.

2. noise in signals

   Everyday sounds contain compared to music or speech, a high amount of noise, resulting from the stochastic behavior of fluids, gases and of collisions of solid objects, friction, squeaks and others. There is a considerable ratio of such *informative* noise as well as non-informative noise from the microphone and other recording conditions.

3. sufficient and general example corpus

   A more practical problem: To give general results and a useful set of tools for measurement and classification the necessary amount of examples of *everyday sounds* is higher than usually used in psychoacoustic studies or otherwise available in the literature. Such a corpus should cover the taxonomy proposed in D4.1 and has to be compiled manually from sound databases.

Low-level spectral features (SLL) give useful information about the sound signal. However, this information is not sufficient to understand sound perception. Thus, psychoacoustically relevant features should be utilized to define efficient representation schemes.

Mel Frequency Cepstrum Coefficients (MFCC's) [27] are well established representation scheme, which dominate applications in speech recognition and music processing. However, for everyday sounds it has been shown [30, 47, 6, 7] that gamma-tone auditory filter-banks yield better classification results than MFCC's.

Therefore, we propose two main directions incorporating biologically relevant features, in order to solve these problems. The first idea is to use time-relative local structures, in order to extract only relevant information from a given sound efficiently. A given sound is decomposed into local features by using time-shiftable kernel functions, which include magnitude a time information as well. The gammatone auditory filters are incorporated as the kernel functions to represent these local features. The gammatone auditory filters are mainly used for identifying the auditory image of a given signal within the cochlea nucleus [37] [38]. A given sound is filtered by the gammatone filters, which simulate the basilar membrane motion for a given sound. However, in our approach, we do not filter a given sound, but try to identify these local features by using the gammatone filters.

The second approach incorporates the gammatone filters in a traditional way. We filter a given sound by a gammatone filter bank, and obtain a multi-channel representation of the given sound. This representation is processed in two different ways to generate feature vectors. The one approach pursues a typical technique, which is used in modulation of signals, namely the Hilbert transform. The second approach is a inner hair model, proposed by Ray Meddis [31] [32]. Inner hair cells are connected to the basilar membrane to transmit the basilar membrane motion to the brain. Therefore this combination simulates the biological processing of a sound in a proper way.

3.1.1 Spike Coding of Audio Signals

Non-stationary and time-relative acoustic features provide useful information about a given sound. Transients, timing relations between acoustic events, periodicities can give clues about the identification and classification of a given sound. However, it is difficult to extract those features. Especially block-based structures cannot detect those features, since these features can appear between two blocks, which neither of these blocks can identify the features properly.

Shift invariant techniques can be incorporated to overcome this problem. However shift invariance alone is not sufficient. A raw signal contains information, which is irrelevant from the identification and classification points of view. Therefore an efficient method to extract only relevant information from the given signal should be developed. In this way, a given sound is represented in a more efficient way as well as the high dimensionality of the problem is reduced enormously.

A sparse, shift invariant representation of sounds [45] overcomes not only the problems encountered by the block-based structures, but also the efficiency of the encoding with respect to the relevance of the coded information is provided as well as the dimensionality of the code is reduced. In this new scheme, a given signal is encoded with a set of kernel functions, which can be located at arbitrary positions in time. The given sounds can then be simply represented as follows:

$$\mathbf{x}(t) = \textstyle\sum_{m=1}^{M} \sum_{n=1}^{n_m} s_i^m \gamma_m (t - \tau_i^m) + \epsilon(t).$$

In this formulation, τ_i^m and s_i^m are the temporal position and the coefficient of the i^{th} instance of the kernel function γ_m, respectively. This representation is based on a fixed number of kernel functions. Each of these functions can be instantiated several times at different locations among the given sound. Hence, the given sound is decomposed into discrete acoustic events, represented by these kernel functions. The kernel functions are selected to be the gammatone filter functions. In other words, a gammatone filter bank is incorporated for identifying these discrete acoustic events. Each discrete acoustic event is defined by three features, namely the temporal position, the amplitude, which is defined as the coefficient of the particular kernel function, and the center frequency of the kernel function (of a gammatone filter). Due to their discrete nature, we call the kernel functions as spikes.

Spike coding is a new scheme to extract only relevant features from a given sound. Thanks to the sparse nature of this scheme, the dimensionality of a given sound is reduced depending on the number of spikes used for representing a given sounds. Since each spike is identified by three features, the total dimension of a spike coded sound is three times the number of spikes used in the code.

The gammatone based spikes are supposed to be biologically relevant. From a physiological point of view, the impulse response of a gammatone filter fits properly well to the impulse response of the basilar membrane from the cat [9]. Psychologically, the frequency selectivity measured physiologically in the cochlea and those measures psychophysically in humans are converging. The latest analysis by Glasberg and Moore [15] shows that the filter bandwidth corresponds to a fixed distance on the basilar membrane. Practically, an nth order gammatone filter can be approximated by cascading n first order gammatone filters. We used the implementation in the auditory toolbox of Malcolm Slaney [44] [43].

Figure 3.1 illustrates one sample sound, an impact sound, and its spike coded representation. In this example a gammatone filter bank of 64 filters used for generating the spike code consisting of 64 spikes. The horizontal axis is the time and the vertical axis is the frequency. Each patch in the bottom figure corresponds to one spike. In this figure, the amplitudes are color-coded. This figure depicts clearly that the spike coding method finds the important parts of a given sound, and codes only these parts efficiently. The method ignores the other parts of the given sound.

In order to find the optimal positions of the spikes, a statistical method called matching pursuit [29] is used.

Matching Pursuit

The matching pursuit algorithm decomposes a given signal into several elements from a given dictionary (the gammatone filter bank). The optimal (spike) element from the dictionary is selected to code a certain portion of the given signal. In this approach, the signal is coded iteratively by these optimal elements by projecting the signal onto each of them. Each filter is convolved with the signal in order to find the optimal filter for the current iteration. The convolution is realized as the scalar product of the dictionary element with the given signal at any location. Hence this can be formulated as follows:

$$\mathbf{x}(t) = \langle \mathbf{x}(t), \gamma_{\mathbf{m}} \rangle \gamma_{\mathbf{m}} + \mathbf{R_x}(t).$$

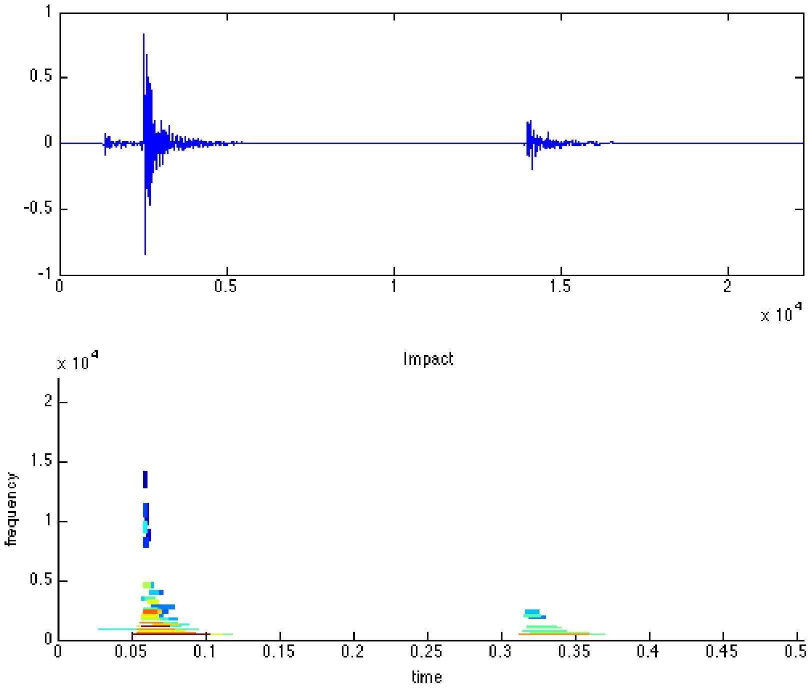

Figure 3.1: An example impact sound and its spike code is shown.

The optimal element of the gammatone filter bank is selected to be the filter with the highest scalar product with the given signal for a certain location:

$$\gamma_{\mathbf{m}} = argmax_m \langle \mathbf{x(t)}, \gamma_{\mathbf{m}} \rangle.$$

The optimal element is subtracted from the original signal at the selected location. In the following iteration, the residual of the signal is considered as the original signal. Hence, the convolution is calculated for the residual:

$$\mathbf{R_x^n}(t) = \langle \mathbf{R_x^n}(t), \gamma_{\mathbf{m}} \rangle \gamma_{\mathbf{m}} + \mathbf{R_x^{n+1}}(t).$$

After each iteration n, the residual of the signal becomes smaller. Two termination conditions can be considered. In the first condition, this procedure is repeated until a certain signal to noise ratio is reached. In this case, the number of spikes used for coding the sounds is different for two given sounds. Another termination condition would be a pre-determined number of spikes for coding. This procedure yields the same number of spikes for each sound, however the signal to noise ratio is different.

A Distance Measure for the Spikes

The spike representation of a sound do not have a vector form. As can also be seen in Figure 3.1, instead of a feature vector, a three dimensional graph representation is generated. The basic idea is to calculate the distance between two spike coded sounds without loosing the three dimensional structure of the representation. In order to preserve this structure a structure preserving distance measure has been defined. This distance measure takes it's values between 0 and 1. The distance between two similar sounds is close to 0. As the similarity decreases the distance value increases and converges to 1.

This distance measure is composed of three individual distances, namely the distance between the amplitudes, frequencies and times. Each of these distances are squared distances. They are defined to be in the interval of 0 and 1. These three distances are weighted differently. These weights can be adjusted depending on the characteristics of the sounds.

$$d = c_t \cdot {d_t}^2 + c_f \cdot {d_f}^2 + c_A \cdot {d_A}^2, \qquad 0 \leq d_{t,f,A} \leq 1$$

In order to have the total spike distance to be between 0 and 1, the weights have to be between 0 and 1. Besides, they have to be chosen so that the sum of c_t, c_f and c_A is 1.

$$\begin{gathered} 0 \leq c_{t,f,A} \leq 1, \\ c_t + c_f + c_A = 1. \end{gathered}$$

Calculating the amplitude difference As we perceive the amplitude of a sound in a nearly logarithmic way, we change the linear amplitude scale of spike coefficients (A) to a logarithmic one A_{lg}).

$$A_{lg} = lg(A).$$

We normalize all the amplitude values in a spike coding. In this way, the relative amplitude differences within a spike coded sound are preserved. However, the normalization enables us to compare two sounds with each other properly. Two similar sounds, one of them being loud and the other being silent, can be identified to be similar by simply ignoring the absolute amplitude differences between the sounds but preserving the relative differences within the sounds.

$$\begin{aligned} A_n &= \frac{A_{lg} - min(A_{lg})}{max(A_{lg}) - min(A_{lg})}, \\ d_A(A_{n,1}, A_{n,2}) &= |A_{n,1} - A_{n,2}|. \end{aligned}$$

Calculating the frequency difference There are two ways to calculate the frequency distance. The first one is to subtract the filter numbers of two spikes and to normalize this value.

$$d_{fb}(f_{b,1}, f_{b,2}) = \tfrac{|b_1 - b_2|}{N_{bands} - 1},$$

where b is the filter number of the spike and N_{bands} is the total number of filters used when performing the spike coding. The advantage of this method is that the frequency difference measure corresponds to the cochlea filter bank that we used for the analysis.

The other way is to have a logarithmic frequency distance measure. After taking the logarithm of the frequency values and normalizing them, we calculate the difference. Apparently this is very similar to the first approach as the cochlea filter bank is almost logarithmically spaced.

$$\begin{aligned} f_{lg} &= lg(f), \\ d_{flg}(f_{lg,1}, f_{lg,2}) &= \frac{|f_{lg,1} - f_{lg,2}|}{lg(f_{max}) - lg(f_{min})}, \end{aligned}$$

where f_{max} and f_{min} are the first respectively the last cochlea band's central frequencies.

Calculating the time difference Time distance is a linear measure. In order to distinguish sounds with a particular time structure from each other, we do not normalize time. For instance, we can distinguish running sounds from walking sounds by comparing the time differences between consecutive foot steps. For these kinds of comparisons, we use time values in seconds. However there are also cases, where relative temporal positions of the onsets within a sound play an important role in the identification of the sound. Opening or closing door sounds can be typical examples for those cases. The time values for those kinds of sounds are normalized.

However for both cases, the time difference is normalized so that the distance values are kept in the $[0, 1]$ interval.

- For the normalized time case, a time value is normalized as follows:

  $$t_n = \frac{t - min(t)}{max(t) - min(t)}.$$

  Thus, the distance will simply be

  $$d_t(t_{n,1}, t_{n,2}) = |t_{n,1} - t_{n,2}|.$$

- For the not normalized time case, the maximum time difference between the spikes of two sounds should be calculated. The maximum time difference can be between the time between the very first spike of the one sound and the very last spike of the other sound.

  $$t_{max} = max(abs(max(t_1) - min(t_2)), abs(max(t_2) - min(t_1)))$$

  Hence, the distance will be

  $$d_t(t_{n,1}, t_{n,2}) = \frac{|t_{n,1} - t_{n,2}|}{t_{max}}.$$

These separate distances between amplitude, frequency and time values of the spikes are summed up to calculate the total distance between two spikes. After calculating the distance between two spikes, the distance between two spike-coded sounds can be calculated as explained in the following section.

A Distance Measure for the Spike-Coded Sounds

The distance between two spike-coded sounds is simply the normalized sum of the distances between the corresponding spikes.

$$d(s^1, s^2) = \frac{1}{N} \sum_{i=1}^{N} d(s_i^1, \mu(s_i^1)),$$

where μ is a mapping between the spikes of the first sound and the second sound, which is used for finding the corresponding spikes to calculate the distance. This mapping is a function defined as

$$s_j^2 = \mu(s_i^1).$$

In order to find the mapping between the spikes of two given sounds a bipartite graph matching algorithm is used.

In order to calculate the distance between two spike-coded sounds, the mapping between the corresponding spikes of these two sounds should be defined. This mapping is defined by using the Hungarian algorithm.

The Hungarian Algorithm

The Hungarian algorithm assigns the vertices of a weighted bipartite graph to each other, so that the maximum weight matching is achieved. In a maximum weight matching the weights must be non-negative.

Definition 1 *A "weighted bipartite graph" is a graph $G = (V = X \cup Y, E = X \times Y)$, such that $X \cap Y = \emptyset$, where an edge $e_{x,y}, x \in X, y \in Y$ has a weight $w(e_{x,y})$.*

In order to achieve minimum weight matching, the highest weight C should be found. Other weights should be subtracted from the maximum weight as follows: $w(e_{x,y}) = C - w(e_{x,y})$.

Suppose that two disjoint sets X and Y of the vertices have the same cardinality n. The weights are shown in an $n \times n$ matrix W. The total weights of the matching M is to be maximized, $w(M) = \sum_{e \in M} w(e)$. A perfect matching is an M, in which every vertex is adjacent to some edge in M. A maximum weighted matching is perfect.

Definition 2 *A "feasible vertex labeling" in G is a real-valued function l on $X \cup Y$ such that for all $x \in X, y \in Y$,*

$$l(x) + l(y) \geq w(e_{e,y}).$$

It is always possible to find a feasible vertex labeling. In order to initialize the algorithm with a feasible vertex labeling, we set all $l(y) = 0$ for $y \in Y$ and for each $x \in X$, we take the maximum value in the corresponding row in the weight matrix, as follows:

$$\begin{aligned} l(x) &= max_{y \in Y} w(e_{x,y}), \\ l(y) &= 0. \end{aligned}$$

If l is a feasible labeling, G_l is the sub-graph of G, which contains only those edges, where $l(x) + l(y) = w(e_{x,y})$. This graph G_l is called the "equality graph" of G for l. A perfect matching M in an equality graph G_l has the following property

$$w(M) = \textstyle\sum_{e \in M} w(e) = \sum_{v \in V} l(v).$$

The main purpose of the Hungarian algorithm is to find the maximum weight matching M in an equality graph G_l, by augmenting the matching M after each step. If augmenting the matching M is not possible, by improving the labeling l to l' a new equality graph $G_{l'}$ can be constructed, which enables to augment the matching M.

In order to improve the labeling, we need to define the neighbors of a single vertex u and a set S of vertices in G_l as follows:

$$\begin{aligned} N_l(u) &= \{v : (u, v) \in G_l\}, \\ N_l(S) &= \cup_{u \in S} N_l(u). \end{aligned}$$

For a set $S \subseteq X$ of the free vertices u and $T = N_l(S) \subset Y$. Set

$$\delta_l = min_{x \in S, y \notin T}\{l(x) + l(y) - w(e_{x,y})\}.$$

and

$$l'(v) = \begin{cases} l(v) - \delta_l & v \in S \\ l(v) + \delta_l & v \in T \\ l(v) & \text{otherwise} \end{cases}$$

The consequences of the relabeling step are

- If $(x, y) \in G_l$ for $x \in S$ and $y \in T$, then $(x, y) \in G_{l'}$.
- If $(x, y) \notin G_l$ for $x \notin S$ and $y \notin T$, then $(x, y) \notin G_{l'}$.
- There will be some edge $e_{x,y} \in G_{l'}$ for for $x \in S$ and $y \notin T$.

Theorem 1 (Kuhn-Munkres) *If l is a feasible vertex labeling and M is a perfect matching for G_l, then M is a maximum weight matching for G.*

The Kuhn-Munkres theorem transforms the problem from an optimization problem into a problem of finding a feasible vertex labeling whose equality graph contains a perfect matching of X to Y.

The Hungarian Algorithm:

1. Start with a feasible vertex labeling l, determine G_l, and choose a matching M in G_l.
2. if M is a perfect matching, then M is maximum weight matching. Stop. Otherwise, there is some unmatched $x \in X$. Set $S = \{x\}$ and $T = \emptyset$.
3. If $N_{G_l}(S) = T$ find $\delta_l = min_{x \in S, y \notin T}\{l(x) + l(y) - w(e_{x,y})\}$.

4. Construct a new labeling l' by

$$l'(v) = \begin{cases} l(v) - \delta_l & v \in S \\ l(v) + \delta_l & v \in T \\ l(v) & \text{otherwise} \end{cases}$$

5. Note that $\delta_l > 0$. Replace l by l' and G_l by $G_{l'}$.

6. If $N_{G_l}(S) \neq T$, choose a vertex $y \in N_{G_l}(S)$ and $y \notin T$.

   a) If y is matched in M, say with $z \in X$, replace $S = S \cup \{z\}$ and $T = T \cup \{y\}$. Go to Step 3.

   b) If y is free, and there will be an alternating path between the free vertices x and y. Construct this path M', and replace M by M'. Go to Step 2.

The Hungarian algorithm provides the complete matching between the spikes of two spike coded sounds. By using this matching μ and the spike distance function (See Section 3.1.1), we can calculate the distance between two given sounds. Being able to calculate the distance between two spike-coded sounds, enables us in turn to generate a distance matrix using these distances. Furthermore, one can define a kernel function departing from the distance measure. In that case, the kernel function can be used in a kernel machine to perform a classification task. In the following, it is explained in detail, how the spike kernel is defined.

The Spike Kernel

By using a kernel function, a non-linear classification problem can be transformed into a linear classification problem in a higher dimensional feature space. In order to do this, a transformation function $\phi(x)$ should be defined, which transforms an input vector into a feature vector in a higher dimensional feature space. The kernel matrix is defined as

$$k(\mathbf{x}, \mathbf{y}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle.$$

Hence, in order to exploit the kernel substitution, a valid kernel function should be defined. A valid kernel function is a function, whose Gram matrix is positive semi-definite.

One approach is to define the transformation function, and by using this function to find the corresponding kernel. Generally, however we don't need to define the transformation function explicitly to construct a valid kernel. One usual way to construct new kernels is to make use of simpler kernel functions as building blocks to define more complicated kernels by simply combining the simple kernel functions in certain ways like addition or multiplication etc. In order to define the spike kernel, we use this combinatorial way.

The linear kernel is defined as

$$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle.$$

The squared distance can be defined by using the linear kernel as follows:

$$||\mathbf{x} - \mathbf{y}||^2 = k(\mathbf{x}, \mathbf{x}) + k(\mathbf{y}, \mathbf{y}) - 2 \cdot k(\mathbf{x}, \mathbf{y}).$$

Since the spike distance is a weighted sum of the squared distances of each of the three attributes of two spikes, we can define the spike distance by using the linear kernel in the same way. The distance between the frequency attributes is calculated as follows:

$$\kappa(f^1, f^2) = k(f^1, f^1) + k(f^2, f^2) - 2 \cdot k(f^1, f^2).$$

In the same way, we can define the distance between the amplitude and time attributes by using the linear kernel as well. In order to complete the spike kernel function, we need to combine these three kernel functions in a weighted sum as follows:

$$d(s^1, s^2) \quad = \quad c_f \cdot \kappa(f^1, f^2) + c_a \cdot \kappa(a^1, a^2) + c_t \cdot \kappa(t^1, t^2). \tag{3.1}$$

So, this kernel function defines the linear spike kernel. Even-though we use the spike distance function in this kernel function, it is still linear in nature. However, we want to define a kernel function, in order to solve non-linear classification problems. Before defining the non-linear spike kernel, we show the Gaussian kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp(\tfrac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2})$$

It is easy to see that the nominator of the exponential function is nothing but the squared distance of the vectors x and y. We can simply replace the squared distance with the linear spike kernel, defined in Equation 3.1 as follows:

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(\tfrac{d(s^1,s^2)}{2\sigma^2})$$

The non-linear spike kernel function is incorporated in the support vector machines to perform classification experiments.

3.1.2 Gammatone Filters based Representation

A gammatone auditory filter bank converts a given signal into a multi-channel basilar membrane motion. Recent research has shown that a gammatone filter can approximate the magnitude characteristics of the human auditory filter properly [11]. Hence a gammatone filter bank simulates the spectral analysis step of the auditory processing of a given sounds performed by the basilar membrane very well. In the following, we introduce two approaches to extend the basilar membrane motion biologically.

The first approach calculates the temporal envelope of each filter output by applying the Hilbert transform. The temporal envelope calculation is an essential part of the roughness estimation proposed by Zwicker and Fastl [53].

The second approach is the inner hair cell simulation by Meddis [31] [32]. The inner hair cells are connected to the basilar membrane. The inner hair cells convert the mechanical transduction performed by the basilar membrane into a neural transduction by simply modeling the transmission rates of each cell into the cleft. A combination of the inner hair model with the gammatone auditory filter bank generates the auditory image of a processed sound.

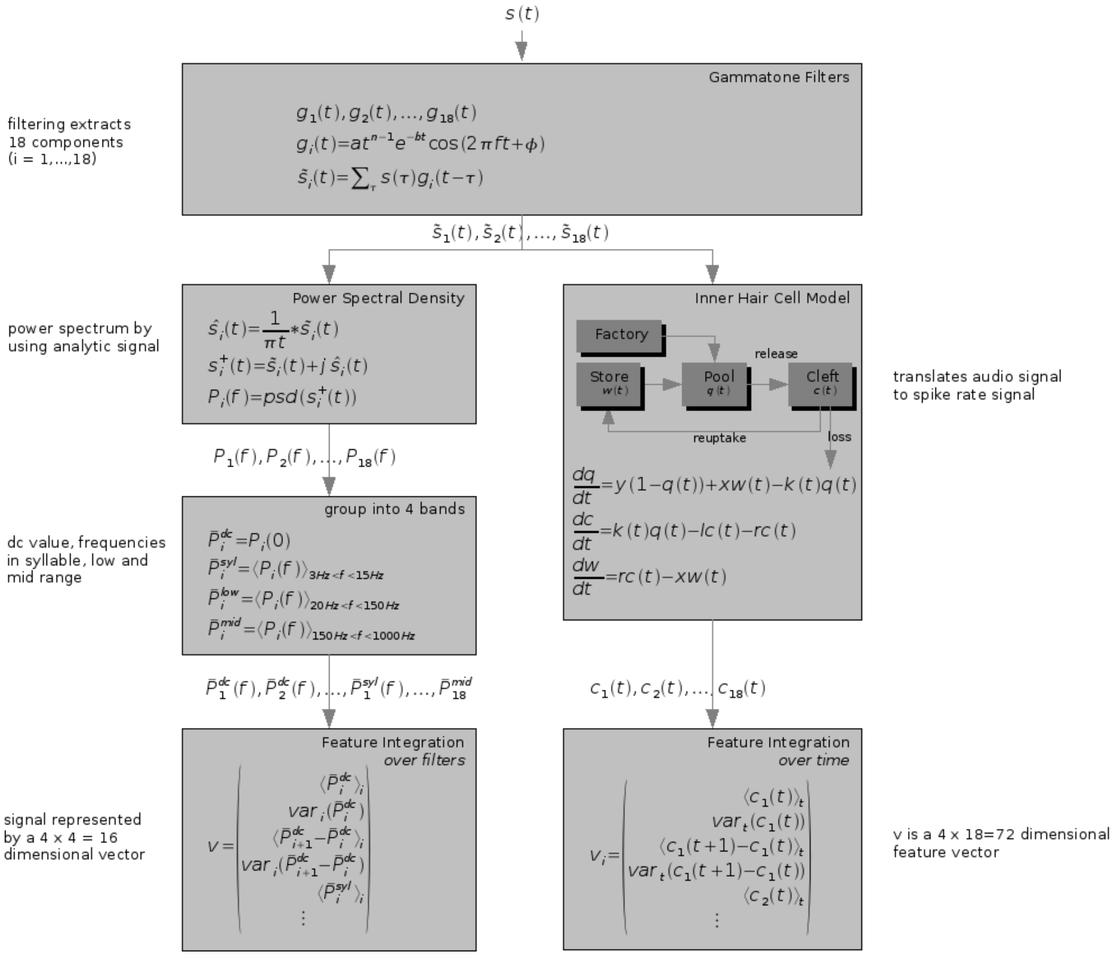
$s(t)$
Gammatone Filters
filtering extracts 18 components (i = 1,...,18)
$g_1(t), g_2(t), \dots, g_{18}(t)$
$g_i(t) = at^{n-1}e^{-bt}\cos(2\pi ft+\phi)$
$\tilde{s}_i(t) = \sum_\tau s(\tau)g_i(t-\tau)$
$\tilde{s}_1(t), \tilde{s}_2(t), \dots, \tilde{s}_{18}(t)$
Power Spectral Density
power spectrum by using analytic signal
$\hat{s}_i(t) = \frac{1}{\pi t} * \tilde{s}_i(t)$
$s_i^+(t) = \tilde{s}_i(t) + j\,\hat{s}_i(t)$
$P_i(f) = psd(s_i^+(t))$
Inner Hair Cell Model
Factory
release
Store $w(t)$
Pool $q(t)$
Cleft $c(t)$
reuptake
loss
translates audio signal to spike rate signal
$\frac{dq}{dt} = y(1-q(t)) + xw(t) - k(t)q(t)$
$\frac{dc}{dt} = k(t)q(t) - lc(t) - rc(t)$
$\frac{dw}{dt} = rc(t) - xw(t)$
$P_1(f), P_2(f), \dots, P_{18}(f)$
group into 4 bands
dc value, frequencies in syllable, low and mid range
$\bar{P}_i^{dc} = P_i(0)$
$\bar{P}_i^{syl} = \langle P_i(f) \rangle_{3Hz<f<15Hz}$
$\bar{P}_i^{low} = \langle P_i(f) \rangle_{20Hz<f<150Hz}$
$\bar{P}_i^{mid} = \langle P_i(f) \rangle_{150Hz<f<1000Hz}$
$\bar{P}_1^{dc}(f), \bar{P}_2^{dc}(f), \dots, \bar{P}_1^{syl}(f), \dots, \bar{P}_{18}^{mid}$
$c_1(t), c_2(t), \dots, c_{18}(t)$
Feature Integration over filters
signal represented by a 4 x 4 = 16 dimensional vector
$v = \begin{pmatrix} \langle \bar{P}_i^{dc} \rangle_i \\ var_i(\bar{P}_i^{dc}) \\ \langle \bar{P}_{i+1}^{dc} - \bar{P}_i^{dc} \rangle_i \\ var_i(\bar{P}_{i+1}^{dc} - \bar{P}_i^{dc}) \\ \langle \bar{P}_i^{syl} \rangle_i \\ \vdots \end{pmatrix}$
Feature Integration over time
$v_i = \begin{pmatrix} \langle c_1(t) \rangle_t \\ var_t(c_1(t)) \\ \langle c_1(t+1) - c_1(t) \rangle_t \\ var_t(c_1(t+1) - c_1(t)) \\ \langle c_2(t) \rangle_t \\ \vdots \end{pmatrix}$
v is a 4 x 18=72 dimensional feature vector

Figure 3.2: Steps of preprocessing. The left side shows the representation that emphasizes the spectral features of the signal. The right one shows the steps using the time domain analysis with the inner hair cell model. In the SVM experiments the whole sound example was used as signal $s(t)$ in opposite to the HMM, for which the signal was windowed and the procedures were applied on each frame resulting in a series of vectors v_i^t.

After performing these two steps, namely applying the gammatone filter bank followed either by the Hilbert transform or by the inner hair cell model, a two-dimensional representation is obtained. In the last step, this representation is integrated efficiently into a feature vector.

Figure 3.2 describes the 2 paths of preprocessing the input signal. Common is the beginning with a gammatone filter bank and the last step to compress a sequence of values into 4 values by using the Mean-Variance feature integration scheme.

Spectral content

Gamma-tone Auditory Filter-bank A gamma-tone auditory filter-bank [38] is the first step of the cochlea simulation, where the basilar membrane motion is simulated in a filter bank. The impulse response of a gamma-tone filter is highly similar to the magnitude characteristics of a human auditory filter, which makes the gamma-tone auditory filter bank a biologically plausible representation. With increasing center frequency, the spacing and the bandwidth of the gamma-tone filters increase, however the overlapping of each consecutive filter stays the same (equivalent rectangular bandwidth or shortly ERB [15]). ERBs are similar to the Bark or the Mel scale.

As a pre-processing of the everyday sounds, we use the gamma-tone filter implementation in Malcolm Slaney's Auditory Toolbox [44]. We use 18 gamma-tone filters in total. Therefore, for each given sound, we obtain 18 filter responses from the gamma-tone filter bank. The lowest center frequency used for the gamma-tone filter bank is $f_{low} = 3\ Hz$. It lies outside the audible frequency range captured by the Basilar membrane. But it captures, in addition, also features that refer to frequencies in the range of roughness or 'rhythmical' content. The highest center frequency used in the gamma-tone filter bank is $f_{high} = 17059\ Hz$ roughly corresponding to the highest audible frequency.

The Gamma-tone filters can be combined with other representations, in order to obtain a more complete representation scheme.

Hilbert Transform The first method with which we combine the gamma-tone filters is the Hilbert Transform [30]. The Hilbert transform of a signal is the convolution of the time domain signal with $\frac{1}{\pi t}$. Combining the Hilbert transformed signal with the original signal, we obtain the analytic signal. This process deletes the negative components of the signal in the frequency domain, and doubles the amplitudes on the positive side. Furthermore the analytic signal is a base band signal. The power spectrum of the analytic signal is the modulation spectrum of the temporal envelope. In many roughness estimations, this step is the very first step of the roughness estimation procedure [53].

Grouping in 4 bands However, the huge dimensionality of the power spectrum of each filter output should be reduced, in order to be able to create a feature vector for a sound. Therefore we summarize these values in four frequency bands [6] [7] by simply taking the average of the power values corresponding to these frequency bands. These frequency bands are the DC values, the frequency interval 3-15 Hz (syllable rate in speech, or rhythms), 20-150 Hz (roughness), and 150-1000Hz (low pass).

Temporal content

Inner Hair Cell Model To analyze the signal in time, we directed the output of the gamma-tone filters into the inner hair cell model of Meddis [31]. In this model, the firing rate of the inner hair cells, connected to the basilar membrane, is modeled. The inner hair cells fire, when a stimulus arrives. This happens when the basilar membrane is deflected at the point of a resonance frequency, where the hair cell sits. This firing is simulated by the dynamics of production and flow of transmitter substance. A certain amount of transmitter substance is released into the synaptic cleft between the hair cell and another neuron, depending on the strength of the stimulus. For each arriving stimulus, the Meddis inner hair cell model calculates these amounts iteratively. In our representation, we use the rate of transmitted part of the transmitter substance [47].

Feature Integration

In both analyses (3.1.2 and 3.1.2) we end up with a number of sequences of values that have to be compressed into one final feature vector. We applied a feature integration scheme that calculates the mean and variance of the sequence as well as the mean and variance of the first derivative (difference of two consecutive values).

In the spectral content case these are 4 sequences (4 bands) of 18 energy values of the filters, which are compressed each to the 4 values described above. This makes a features vector of 16 values. Note, that the derivatives (deltas) were taken over the the filter outputs, i.e. in frequency domain.

In the temporal case we used the same schema in the time domain. Here we had 18 time sequences from the Inner Hair Cell model, which were compressed each to 4 mean/variance values resulting in a 72 dimensional vector.

3.1.3 Other Representation Schemes

In the sequel, we will use other representations of sounds as a comparison.

Mel Frequency Cepstrum Coefficients (MFCC's) [27] are a well established representation scheme which dominates applications in speech recognition and music processing. It is based on a frequency spacing (Mel scale) inspired by the basilar membrane. This representation allows to separate a signal into source (fundamental frequency) and filter (spectral shape) component. We use 13 MFCC coefficients, their variances, finite differences between consecutive MFCC coefficients and the variances of these differences, adding up to a 52-dimensional vector. MFCCs have rarely been applied to environmental sounds. We will use them in Section 3.2.2.

As a comparison, we will use a feature set we will call Low-level spectral features (SLL). It consists of a 68-dimensional vector: the 13 MFCCs and four additional descriptors (signal energy, centroid, roll-off, zero crossings) their variances, finite differences and the variances of the differences. They will also be used in Section 3.2.2.

A large set of features have been implemented in the Ircam descriptor by G. Peeters [39]. In Section 3.2.2, we will also use a feature set of timbre descriptors. It is a 19-dimensional feature that consists of mean and variance for spectral sharpness and spread, and 12 dimensional signal auto correlation, roughness, energy modulation frequency, and energy modulation amplitude. Sharpness is the perceptual equivalent to the spectral centroid. Spectral spread measures the distance between the largest to the total loudness value. Roughness, how it is used here, is based on the ERB scale.

The synthesized impact sounds in Section 4.4 will be represented by the following features. We will use unnormed timbre features (42 dimensions) consisting of: 6 dimensions perceptual spectral centroid mean, 6 dimensions perceptual spectral spread mean, mean energy, energy modulation frequency, effective duration, temporal decrease, temporal centroid, loudness mean, relative specific loudness mean.

In Section 4.4, we will represent car horn sounds by the three dimensions: fundamental frequency, roughness, and spectral spread.

3.2 Evaluation

3.2.1 Spike Representation

In order to show that this new approach works generally well for simple everyday sounds, we performed two experiments on two simple data sets. The first data set contains footstep sounds of running vs. walking people on different floor types. In order that the sounds are homogeneous, we prepared sounds of 10 footsteps each. Figure 3.3 shows one running and one walking sounds. Both of these sounds are spike coded, where time component is not normalized, simply because running and walking can easily be distinguished by comparing the time between consecutive footsteps.

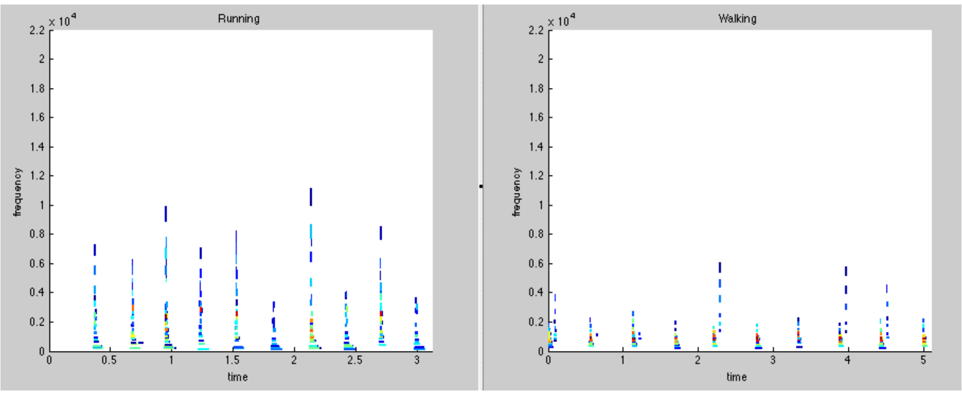

Figure 3.3: One spike coded running and one spike coded walking sound are shown in comparison.

The second data set contains single footstep sounds on two different floor types, namely on leaves and on marble. In this experiment, we normalized the time component in the spike coding. Figure 3.4 shows two spike coded sound examples, one footstep one marble and one footstep on leaves.

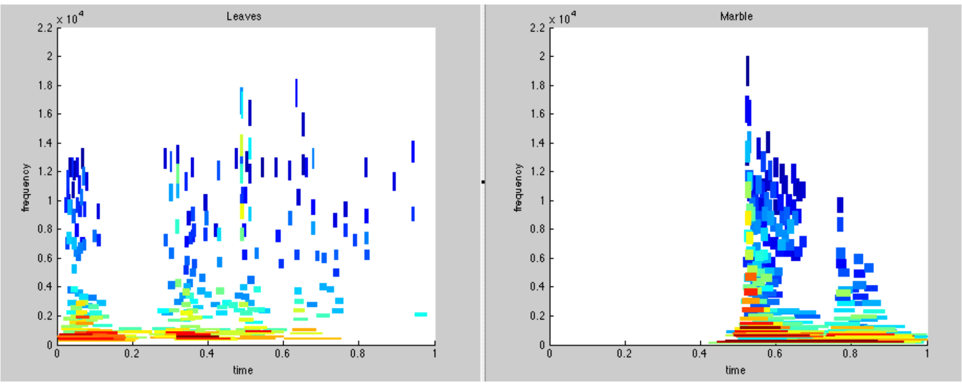

Figure 3.4: One spike coded footstep sound on leaves and one on marble are shown in comparison.

These two data sets were tested by using SVM's in binary classification scenarios. The distance matrices of these two experiments are shown in Figures 3.5 and 3.6. In both of these figures, the sounds are sorted within their class labels. Therefore it is easy to observe the clusters for each class. The upper left and bottom right regions of the distance matrices define clusters for each class, where the distances between two sounds from the same sound class are significantly lower than the distances between the sounds, which belong to different classes.

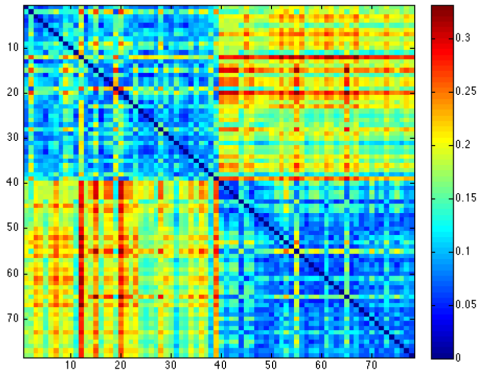

Figure 3.5: The distance matrix of the spike coded running and walking sounds is shown.

Both of these experiments yielded 100% of accuracy. The experiments were performed in a leave-one-out (CVLOO) cross validation manner, where we left one sound out of the training set and performed the test only for this single sound. The results are average of each of these CVLOO cross validation runs. Hence, the method could predict the label of each of these sounds correctly.

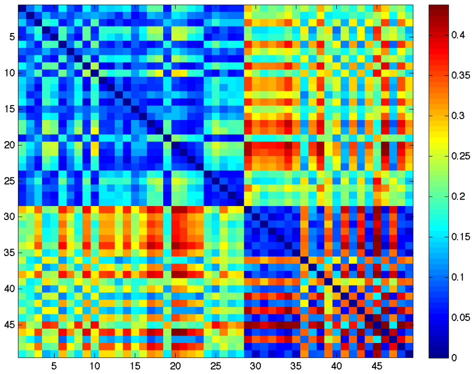

Figure 3.6: The distance matrix of the spike coded footsteps on leaves and on marble is shown.

3.2.2 Representation Comparison

For the experiments, recordings of footsteps and doors are selected from the sound collection "Sound Ideas" [22].

The footsteps are a subset of the Foley collection. It consists of recordings of different kinds of shoes (heels, boots, barefoot, sneakers, leather) on various grounds (concrete, wood, dirt, metal, snow, sand). We picked the following movement modes: walking, running, jogging from male and female subjects. The recordings are cut, when necessary, such that exactly one step is contained in one sample. For the experiments on concrete and wood floor (Tables 3.2, 3.3) we used 44 files per class and in the last 6-class experiment we used 40 files per class (Table 3.4).

We performed experiments on different sole types, namely barefoot, sneakers, leather, heels and boots. These sole types were classified on two different floor types, namely on concrete and wood floor. Furthermore, we performed an experiment to classify six floor types ignoring the sole types. These floor types were snow, sand, metal, dirt, wood, concrete.

The labels of these data-sets are mostly psycho-acoustically validated. We did not perform detailed psycho-acoustical experiments, but we checked the sounds by listening to them by ourselves. We discarded sounds whose class cannot be identified while listening to them.

In order to evaluate the results of our experiments with the gammatone based representation schemes, MFCC, low level and timbre descriptors (cf. Section 3.1.3), we used leave-one-out cross validation, where all sounds but one are incorporated into the training set. Then the trained algorithms are tested on the remaining single sample that previously had been excluded from the training set. This procedure is repeated for all possible partitions into training/test sets.

The results of the experiments opening vs. closing door sounds are shown in Table 3.1.

The table shows the results of SVM and HMM classifications using two different representations: 1) gamma-tone filters combined with the Hilbert transform (GT Hil), 2) gamma-tone filters combined with the inner hair cell model (GT Med).

SVM		HMM	
GT Hil	GT Med	GT Hil	GT Med
84.0%	63.2%	57.2%	70.0%

Table 3.1: Classification of opening vs. closing door sounds

We performed multi-class experiments for footstep sounds, where we used five different sole types on concrete floor, performing five separate binary classification runs. In each run, one class is classified against the rest, which consisted of the other 4 classes. Therefore the rest-class constituted a heterogeneous mixture of sounds. We took the average of these five binary classifications. Table 3.2 shows not only the overall average of the separate classification experiments, but also the results of the separate binary classification experiments.

The percentages are 100% - BER (balanced errors rate), thats weights the two class errors equally. First we look at classification of shoe types on concrete.

	SVM					HMM	
	MFCC	SLL	TD	GT Hil	GT Med	GT Hil	GT Med
Barefoot	94.4%	94.4%	98.6%	97.7%	**99.7**%	85.2%	94.0%
Sneakers	76.6%	82.1%	76.1%	**90.1**%	83.0%	73.2%	74.6%
Leather	**100**%	**100**%	91.8%	90.1%	92.6%	90.0%	92.0%
Heels	**100**%	**100**%	98.8%	95.9%	97.1%	88.0%	89.4%
Boots	70.0%	72.5%	86.1%	**94.3**%	92.1%	89.9%	84.6%
Average	88.2%	89.8%	90.3%	**93.6**%	92.9%	85.3%	86.9%

Table 3.2: Experiment 1: Classification of the sole types on concrete floor

The same sole types hitting a wooden floor were classified in experiment 2. Following the same criteria as explained before. The results for this experiment are shown in Table 3.3. Note that the Sneakers class yielded the lowest accuracy, decreasing dramatically this result for gamma-tone Meddis based representation. A particular feature of Sneakers is their noisy characteristics compared to the other types of shoes.

In the third experiment (Table 3.4) we classified different floor types: snow, sand, metal, dirt, wood, concrete. The gamma-tone based representations outperformed the other methods. Dirt is not classified as well. This could be linked to its more complex temporal structure and its rather inhomogeneous sound character. Note that the timbre descriptors chosen are not suitable in this classification task. However, as future work, further exploration of new combinations of timbre descriptors [39] is needed.

Except in one case SVM performed better than HMM. For shoe type, on average the power spectrum based method (left in Figure 3.2) performed better than the Meddis based right method. However for the floor types the Meddis based method clearly

	SVM					HMM	
	MFCC	SLL	TD	GT Hil	GT Med	GT Hil	GT Med
Barefoot	83.1%	**95.9**%	92.2%	95.6%	**95.9**%	88.0%	84.3%
Sneakers	81.7%	83.7%	76.5%	**87.5**%	75.9%	69.8%	67.2%
Leather	93%	97.7%	92.7%	**98.5**%	94.8%	91.6%	89.8%
Heels	99.7%	99.7%	97.7%	98.5%	**100**%	98.8%	99.4%
Boots	95.9%	**96.5**%	90.1%	96.2%	92.7%	88.1%	92.7%
Average	90.7%	94.7%	89.8%	**95.3**%	91.9%	87.3%	86.7%

Table 3.3: Experiment 2: Classification of the sole types on wood floor

	SVM					HMM	
	MFCC	SLL	TD	GT Hil	GT Med	GT Hil	GT Med
Snow	96.5%	98.7%	93.3%	90.0%	**99.5**%	94.0%	89.9%
Sand	90.8%	**98.5**%	86.3%	86.4%	91.0%	88.2%	92.8%
Metal	97.5%	98.5%	85.8%	86.2%	**98.8**%	89.2%	92.9%
Dirt	61.7%	82.5%	72.8%	77.7%	**90.0**%	77.5%	83.1%
Wood	96.0%	98.8%	87.8%	98.3%	**100.0**%	89.4%	94.8%
Concrete	88.7%	88.9%	82.5%	91.5%	**98.0**%	92.5%	89.3%
Average	88.5%	94.3%	84.7%	88.4%	**96.2**%	88.5%	90.5%

Table 3.4: Experiment 3: Classification accuracy of the floor types

performs better.

3.2.3 Outlook: General Everyday-Sounds Classification

In order to generalize the prediction performances of the representation schemes we provide, we need to define a general sound database corpus according to the results obtained by the soundscape and categorization studies on everyday sounds.

Vanderveer's categorisation experiments [49] revealed that human beings grouped sounds together either because they were similar or because they were produced by similar events. Different levels of abstraction are used, when grouping sounds into a sound category. These abstraction levels are used in defining the categories or even a complete taxonomy as well. Gaver [14] proposed a hierarchical taxonomy of sound events depending on the information about an interaction of materials at a location in an environment. He claims that a sound event occurs due to the interaction of two materials. Therefore, Gaver proposed a general level in this hierarchy depending on the general categorisation of these materials, namely solid, liquid and gas. Univerona made use of the taxonomy to develop their physically-based sound models, and in turn extended the taxonomy depending on the sound-producing events. Even-though the taxonomy developed by Univerona depends basically on Gaver's taxonomy, there are differences between them depending on the low-level models developed by Univerona used for creating basic sound events.

Gaver defined the sub-categories within the solid sounds as deformation, impact, scraping and rolling sounds. Univerona defines fracture, impact and friction as the low-level models generating vibrating solid sounds. Among the two categorisation schemes, only the impact sounds are common. Scraping and friction can also be considered as similar. Univerona defined rolling sounds as a compound category consisting of infinitely many impact sounds. Hence impact model generates also the rolling sounds. In the Univerona taxonomy, deformations like crushing and breaking are defined as compound categories consisting of fracturing and impact sounds. Vacuum cleaners, balloons, steam and wind sounds are categorised in the wind sounds.

In Gaver's taxonomy, liquid sounds are categorised as drip, pour, splash and ripple, whereas Univerona has only two sub-categories, namely bubble and flow. In Univerona's taxonomy, dripping is a combination of bubble sounds with impact sounds from the solid sounds category. Pouring is even one level higher in the hierarchy. Univerona defines pouring as a combiantion of dripping and flowing. Splashing is a complex form of dripping. Hence, splashing is one level higher in the hierarchy as well. Gaver divides the gas sounds into three sub-categories: explosion, whoosh and wind. Univerona, on the other hand, has only two categories. These two categories are turbulance and explosion. So, explosion category is common in these two models. In Univerona's taxonomy, turbulance model generates whooshing sounds.

The Gaver taxonomy is an ecological approach depending on the assumption that sounds provide information about the interaction of the materials. The taxonomy proposed by Univerona is formed based on the Gaver taxonomy depending on the low-level sound models, generating everyday sounds. In this respect, the Univerona taxonomy can be considered as a realisation of the taxonomy proposed by Gaver. There are other taxonomy schemes categorising everyday sounds into basic and compound sound classes considering the perceptual issues as well.

We generate our general everyday sounds database, considering these two taxonomy schemes in combination. We selected the sounds from the Sound Ideas database. The semantic descriptors of the Sound Ideas database are used for labelling the sounds. We used a hierarchical categorisation as well. In the lowest level of this hierarchy, we divided our sound database into three main sound categories, namely solid, liquid and gas.

The solid sounds category consists of four sub-categories. These are impact, rolling, deformation and friction sounds. Impact sounds are selected to be switches, single typewriter clicks and hitting sounds. The rolling sounds are selected to be the car engine sounds in the idle state. Gaver indicates the regularity in the rolling sounds. Car engines in their idle state make regular rolling sounds. Therefore these sounds can represent the rolling sound category properly. We simply cut 4 sec. long snippets of the car engine sounds. Car crashes and glass crashes sounds constitute the deformation sounds. The friction sounds category consists of squeaking and sliding doors, windows and dragging sounds. The liquid sounds are sub-divided as pouring, dripping and flowing sounds. Pouring sounds category consists only of pouring sounds. However the dripping sounds category constitutes dripping, splashing sounds as well as rain and waterfalls. Simple dripping sounds are quite different than rain or waterfall sounds, but in the taxonomies we refer to, they are considered as dripping sounds as well. In the Univerona taxonomy, pouring sounds are considered as a combination of dripping and flowing, but we stick to the Gaver taxonomy in this case, and distinguish pouring sounds from dripping sounds. Filling, waves coming in and running (flowing) water sounds are grouped in the flowing sounds category. Considering the fact that pouring sounds generally fill a bottle or a can with running liquids, pouring and flowing sounds can be grouped together as well, which is the case in the taxonomy of Univerona. The sub-categories in the category of gas sounds are explosions, wind and whooshing sounds. Explosions and gunshots sounds are considered as explosions. Passing airplanes or cars, flame thrower sounds, fire burst sounds, arrows and knifes are grouped in the whooshing sounds category. The wind sounds category consists of wind, vacuum cleaner sounds, air, balloon and steam sounds. The vacuum cleaner sounds generate not only wind sounds but also rolling engine sounds. So, they can be considered in both of these categories.

This sound database corpus will be classified by using our representation methods. These classification experiments will provide us the generalizability of our representation methods onto each of these everyday sound categories. Furthermore, the classification results will be analyzed further, in order to extract the perceptually relevant features, which play an important role in the classification.

4 Case study: Impact Sounds

The aim of this case study is to achieve Milestone M1: Analysis, classification and measurement of functional sounds. It is based on findings from WP4 (context information) and WP2 (physical modeling parameters).

The collision of solid objects generates sound that is a basic building block for many *everyday sounds*. Either as a single event or as a composition. The information conveyed by the sound of colliding solids is manifold: weight, material, used force, shape and others (see and considerations and literature review in D4.1 [19] section 1.1.2). In this case study we focused on the perception of the material of hammer struck solids.

This chapter is organized in two separate studies: 1. Learning how to control an impact sound generating process using perceptual information from subjects. 2. Using the spike representation for impact sounds for automatic classification to perceptual categories.

Both studies use the data acquired by our partners at IRCAM (WP4) using a physical sound model by UNIVERONA (WP2).

4.1 Perceptual control of Impact Sound Synthesis

Developing tools for sound design requires a notion about the perceptive function of synthesized sound. This is provided by the designer. For the design process it is important to take into account what information a subject (a user of product) perceives when hearing a functional sound.

A direct examination of the relation between sound synthesis and perception was done using a sound synthesis model that is controlled by set of numerical parameters. The synthesized sound was then played back to subjects that assigned it to one of four possible material classes. Using this data a human perception model was trained.

4.1.1 Simulating Materials

Modal synthesis [1] is a physically based sound synthesis technique that is able to produce realistic sounds emitted by vibrating solid objects. It simulates oscillations effected by an excitor by superposition of a number of damped resonators. In this study we used a model with a hammer and a resonator, modeled by a second-order linear oscillator system:

$$\begin{aligned}
\ddot{x}_i^{(h)} + g_i^{(h)}\dot{x}_i^{(h)} + \left[\omega_i^{(h)}\right]^2 x_i^{(h)} &= \frac{1}{m_{il}^{(h)}}(f_e^{(h)} + f)\,,\; i = 1\dots N^{(h)} \\
\ddot{x}_j^{(r)} + g_j^{(r)}\dot{x}_j^{(r)} + \left[\omega_j^{(r)}\right]^2 x_j^{(r)} &= \frac{1}{m_{jm}^{(r)}}(f_e^{(r)} + f)\,,\; j = 1\dots N^{(r)} \\
x &= \sum_{j=1}^{N^{(r)}} t_{mj}^{(r)} x_j^{(r)} - \sum_{i=1}^{N^{(h)}} t_{li}^{(h)} x_i^{(h)} \\
v &= \sum_{j=1}^{N^{(r)}} t_{mj}^{(r)} \dot{x}_j^{(r)} - \sum_{i=1}^{N^{(h)}} t_{li}^{(h)} \dot{x}_i^{(h)}
\end{aligned}$$

$$f(x,v) = \begin{cases} kx(t)^\beta + \lambda x(t)^\beta \cdot v(t) & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases} \text{(impact force)},$$

where x is the oscillator displacement, f an external driving force. Index (h) denotes the excitor (hammer) and (r) the resonator (hit object). The two oscillator systems are parameterized each by g the damping coefficient, ω the oscillator center frequency and m_{ij} the mass of the object at a given point (i,j). For any point (l,m) the resulting displacement x is calculated using the matrix $\mathbf{T}$ that decouples the oscillator equations [1]. The last equation describes the driving force $f(x,v)$ that couples the hammer and resonator. For more details on this see also [21] and [40].

Modal synthesis is specialized on solid compact objects rather than modeling vibrating air in tubes or strings of instruments. Because we are interested in everyday sounds it is the right tool to study them as long as we are dealing with solids like shoes, wheels, doors or other mechanical devices. A large fraction of everyday sounds is covered.

Hitting solid objects emits sounds that convey information about the material of both the excitor and the resonator. We confine ourselves to look at the material of the resonator only, while keeping the excitor fixed for the sake of simplicity.

Referring to the equations of the modal synthesis above, one can count for each mode 3 parameters (g_i, ω_i, m_{il}) for hammer and resonator plus the 3 coupling parameters k, λ and β. Given a 2-modal system used for resonator and a fixed setting for the hammer there are 9 parameters to adjust. Tests show this is enough to simulate *cartoonified* impact sounds of the resonator materials glass, metal, plastic and wood.

The system is a transformation of a parameter vector $\theta = (g_1, g_2, \omega_1, \omega_2, m_1, m_2, k, \lambda, \beta)^T$ to a a signal $s(t)$, which is listened and categorized by a subject according to the perceived material

$$\mathfrak{m} \in \{\mathfrak{C}_{\text{glass}}, \mathfrak{C}_{\text{metal}}, \mathfrak{C}_{\text{plastic}}, \mathfrak{C}_{\text{wood}}\},$$

where $\mathfrak{C}_x$ is the label number of the material class x (see top part of figure 4.1).

The key question we want to address here is: How can we produce material sounds efficiently to meet the perceptive expectation of a certain material?

4.1.2 Prediction of Material Classes

The link between material perception and synthesis parameter settings and the signal is still missing. In particular we are looking for the mapping $h : \theta \mapsto \mathfrak{m}$ predicting the material from the input parameters (see also figure 4.1). h can be applied as a sound design tool to give feedback to the designer.

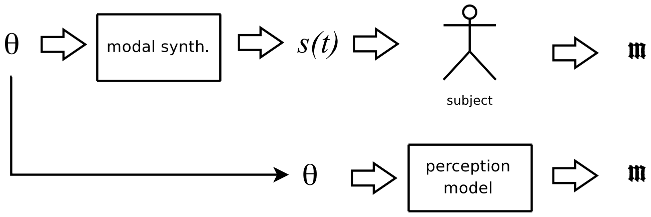

Figure 4.1: models of perception

To solve the task of predicting materials we use methods of machine learning using examples, that can be obtained from psychoacoustic experiments (section 4.1.3). First, let

$$\mathfrak{m} \in \{-1, 1\}$$

be a binary classification task. We seek a classification function:

$$h(\mathbf{x}_i) = \mathrm{sign}(\langle \varphi(\mathbf{x}_i), \mathbf{w} \rangle + b) = y_i$$

$\varphi(\mathbf{x}_i)$ being a feature vector ($= \varphi(\theta)$) and y_i the prediction of $\mathfrak{m}$ given $\mathbf{x}_i$. Either side of the discrimination border $\langle \varphi(\mathbf{x}_i), \mathbf{w} \rangle + b$ belongs to one of the two classes. K is always a linear or non-linear Kernel function (compare section 2.2.3) with $\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle = K(\mathbf{x}, \mathbf{x}')$. Finding $\mathbf{w}$ and b such that the model will predict correctly the example data while generalizing (minimize the generalization error) is the task to solve.

The advantage to use the modal synthesis rather than recordings is, that we are able to produce a wide spectrum by sampling the parameter space to cover it equally under similar conditions and thus can infer a general rule for solid sounds.

The parameters of the modal synthesis can be used directly as features of the impact sound. These features are *generative*, since they are not calculated from the signal, but are used to generate it. Still, they are sound descriptors that, given the synthesis process, determine the sound and the material.

We have to deal with the fact, that not all possible parameter vectors generate necessarily a material sound, not even an audible sound. To generate feasible examples to train a prediction model it is necessary to filter out those non-material sounds. In a dynamic system that can show chaotic behavior there is no direct way to do this by analyzing the parameters. Thus, in the first stage of the psycho-acoustic experiments a manually selected set of parameters vectors was used, which were validated by a human to be a *good* sound. The size of this manually selected set is limited though, in the second stage we plan to use a larger one that needs to be generated automatically.

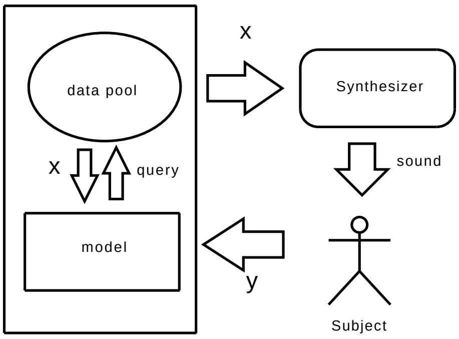

Figure 4.2: Setup of a psychoacoustic experiment using *Active Learning*

4.1.3 Experiments

The experimental setup is shown in figure 4.2. We used active learning (section 2.2.4) to optimize the learning progress. The synthesis model used is governed by 9 parameters (section 4.1.1). We chose 6 of them that influence the generated sound significantly. These are the parameters of the 2 resonators. The 6 parameters span the input space. We are looking for the 4 regions in that space that belong to the materials.

The approach of quantisizing the input space and let these examples be labeled by subjects fails for 2 reasons: (1) The 6 dimensions cannot be densely sampled[1] and (2) the mapping $\Theta \rightsquigarrow \mathfrak{M}$ is only partially defined: not all parameter combinations induce a clear material sound. Either the sound is too synthetic or not audible. Therefore, a small set of sounds was chosen manually [28] by experimenting with the sound model. This yields a complete surjective mapping $\Theta' \rightarrow \mathfrak{M}$, where $\Theta' = \{\theta_1, \theta_2, \ldots, \theta_n\}$ is a discrete set of parameter vectors [2].

The experiments were done using the two sets: Θ' and $\Theta'_{\mathrm{red}} \subset \Theta'$ with a size of 372 and 196 examples respectively, since the subjects labeled the same sounds differently. A small analysis revealed that none of these sounds have been labeled by the subjects the same. These differences caused inconsistencies within the data set. In order to get rid of these inconsistencies, IRCAM generated a sub-set of the same data set from consistently labeled sounds. The reduced set is a more restricted selection in terms of consistency between the subjects. The examples with the most disagreement were removed and considered to be ambiguous.

In the complete data set experiments, we did binary and four-class classification experiments with 372 impact sounds. These four classes are the material classes, namely wood, plastic, glass and metal. In the binary experiments, wood and plastic as well as glass and metal classes are summarized in one class. These sounds have been labeled by 20 subjects in psycho-acoustics experiments, performed by IRCAM. We simulated

[1] 10 steps per dimension would 1 million sample points

[2] Instead of manually selecting examples it is planned to use an automatic selection process that yield a much larger set Θ'' to choose from. To get good examples we would need again a predictor that can distinguish between material and non-material sound, see section 4.1.5.

each subject separately. Therefore we obtained 20 different classification results. Each of these results can be considered as the simulation of one particular subject.

Generally the accuracy of the results are satisfactory. However, for some subjects, namely 6, 7, 8, 10, the binary classification results are worse than the four class classification. Note, that not all available label information was used, but the training was stopped after 30 labels. This was done to simulate that in a psychophysical experiments the number usable labels will always limited.

4.1.4 Results

Figure 4.3 summarizes the the results of the predictors trained with active learning as boxplots (see figures 4.6 and 4.7 for detailed results). The training algorithm used 20 labeled examples and 252 unlabeled examples. The rest of 100 examples were used for testing, from which the resulting prediction error were calculated for each subject.

The 4 class experiment shows a much higher error rate than the 2 class experiment. This is due to 2 reasons: A *guessing* predictor in a 4-class task would converge to a rate of 75%, while a binary one would go to 50%. Secondly, the task is in general harder for four classes, since there the 4 class predictor, that is constructed of 4 binary ones, has to find a model that suites all classes evenly well. Still, even the 4 class prediction here is clearly better than guessing and the binary task is fairly good, taken into account that is has only seen 20 examples. The reduction of label noise in the reduced data set helps training as expected in both prediction tasks.

4.1.5 Discussion

For the data given, the results are very promising. Nevertheless for practical application we need a significant improvement. This can be achieved by using a much larger pool of unlabeled data. Acquiring more data points that can be used for training is possible:

1. Having a method to decide whether a sound is a material sound or not would give us a possibility to sort out those points that are not of interest. Psychoacoustic measures for loudness and can help here.

2. Introduction of a 5th negative class: When a sound is played the subject can decide to label it *undecided* or *not audible*. This way one could use the physical model directly in the experiment. The crucial point here is how much is the ratio of non-material sounds to material sounds. If there are much more non-material sounds – as it seems to be – the training set would have be very unbalanced and training would take longer than a subject can do it. Furthermore subjects are going to label sounds in-between two classes (metal/glass) as *undecided* and lower the amount of positive examples. Studies on forced-choice decision making will have to be consulted.

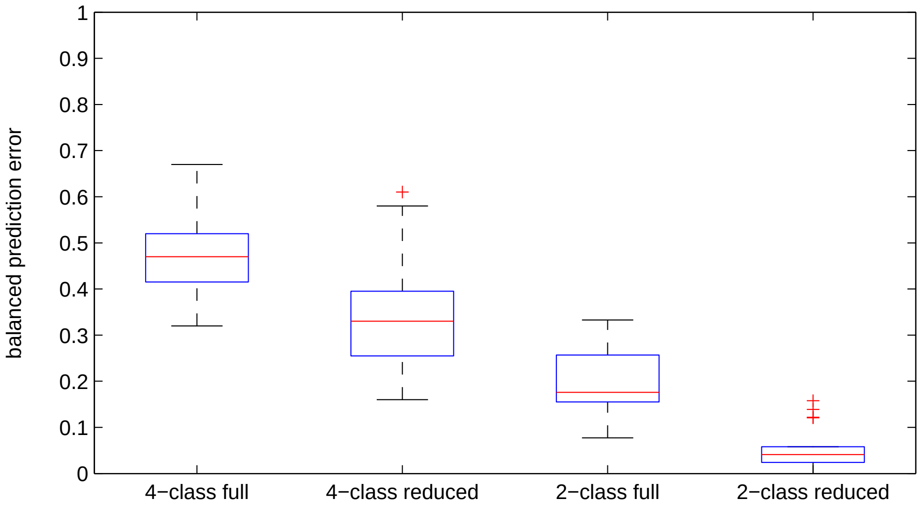

Figure 4.3: Prediction error after training with modal synthesis parameters. Each experiment simulation was carried out with 20 subjects. The test results are shown as boxplots of the 20 trained human prediction models. The center red line is the median error rate, whereas the box covers 50 % of the result distribution. See detailed results at the end of this chapter.

4.2 Representation of Impact Sounds

The input parameters θ of the physically based sound model determine the sounds, which have been synthesized and saved as audio files (samples) in order to generate a feature-based representation of the signal. For these sounds we generated spike code representations using a gammatone filter bank with 256 filters. The number of spikes was fixed to 16 spikes per sound example. The representation method was presented in section 3.1.1. Since these impact sounds are relatively short, only 16 spikes are sufficient to code the perceptually relevant parts of these sounds.

In these experiments, similar to the experiments presented in Section 4.1.3, each subject is modeled separately. We modeled four class experiments as well as binary classification experiments. In these experiments, the complete data set and the reduced data set are used.

4.2.1 Results

The results of the complete as well as the reduced data sets are shown in Figure 4.3. This figure indicates better results than the results obtained by using the physical model parameters. The input parameters of the physical model are generative parameters to synthesize these sounds. However generative parameters do not necessarily reflect the

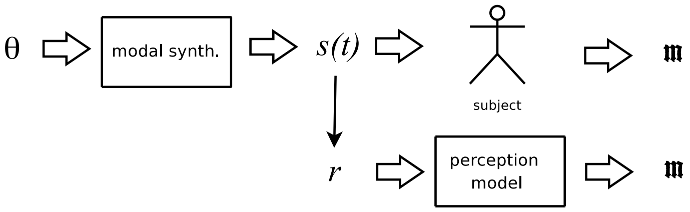

Figure 4.4: The sound signal $s(t)$ is represented by $\mathbf{r}$, which is used to train the perception model.

perceptual features, which play an important role in the classification of these sounds. However the spike coding representation scheme is based on biological and psychophysical studies, where the perceptual behavior has been investigated thoroughly. Therefore this representation scheme encodes the perceptually relevant features in classification in a better way than the physical model parameters. The results we obtained by these experiments also encourage us to claim that the spike coding combined with the Hungarian algorithm is a proper way to encode given sounds from a classification point of view.

The results are shown in figure 4.5. As expected the results obtained with the reduced data set are far better than the results obtained with the complete data set.

4.3 Conclusion

From the results of these experiments with synthesized impact sounds we conclude:

1. The division of the control parameter space into 4 regions was possible, but we have to accept a considerable error. This shows that the selected generative features and the perception of materials is not straight forward. This has to be considered when building tools. Solely using control parameters with human labeling as measurement is not sufficient.

2. The division of the feature space using a spike representation with a kernel that measures similarity between spike patterns yielded good results. We could achieve prediction rates that are feasible to use. Here the the perceptual information about the material is available and the feature space is flexible enough to allow successful classification depending on the notion about materials of the subject, which can differ significantly.

4.4 Outlook: Preliminary Experiments with Timbre Descriptors

In order to further understand the nature of spike coding in a better way, and to study its performance comparatively, a comparison with timbre descriptors will be fruitful. Moreover, the timbre descriptors can be compared with the descriptor built from the gamma-tone filter bank and subsequent inner hair cell model (GT Med), for the same

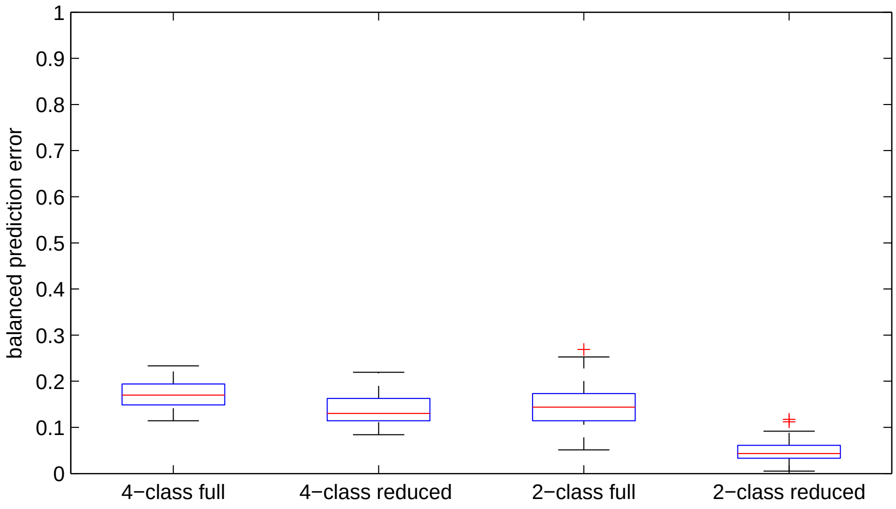

Figure 4.5: Prediction error after training with spike coded impact sounds. Each experiment simulation was carried out with 20 subjects. The test results are shown as boxplots of the 20 trained human prediction models. The center red line is the median error rate, whereas the box covers 50 % of the result distribution. See detailed results at the end of this chapter.

reason. These comparison experiments should be performed on the entire group of subjects separately. These results will enable us to gain insights of the psychoacoustical relevance of the representation schemes properly.

4.4.1 Car Horn Sounds

In another preliminary experiment we compared 41 sounds. Subjects have been asked to judge whether the sounds are car horns or not [25]. According to the average answers of 31 subjects the sounds are grouped into car horn and non-car horn sounds. With support vector machine, the leave-one-out procedure, and the grid search method we yield a balanced recognition rate of 73.56% when using the GT Med representation and 84.93% when using a 3-dimensional representation consisting of fundamental frequency, roughness, and spectral centroid. However, a sound corpus of only 41 sounds cannot yield significant results about a high-dimensional representation scheme like the GT Med representation. In order to draw meaningful conclusions about these kinds of high-dimensional representation schemes, they should be evaluated by using larger data sets. Therefore we will redo these experiments using data sets consisting of sufficient number of sounds.

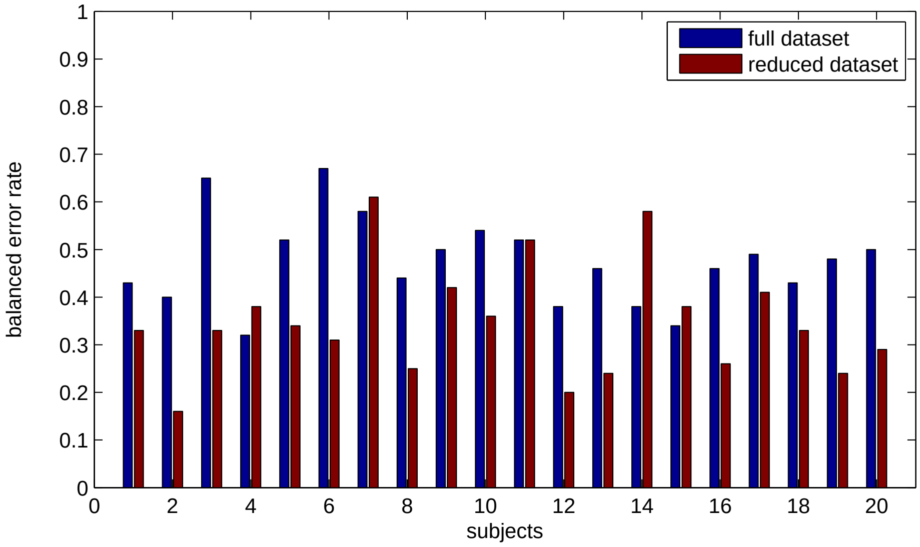

Figure 4.6: Results 4-class prediction with Active Learning using synthesis parameters

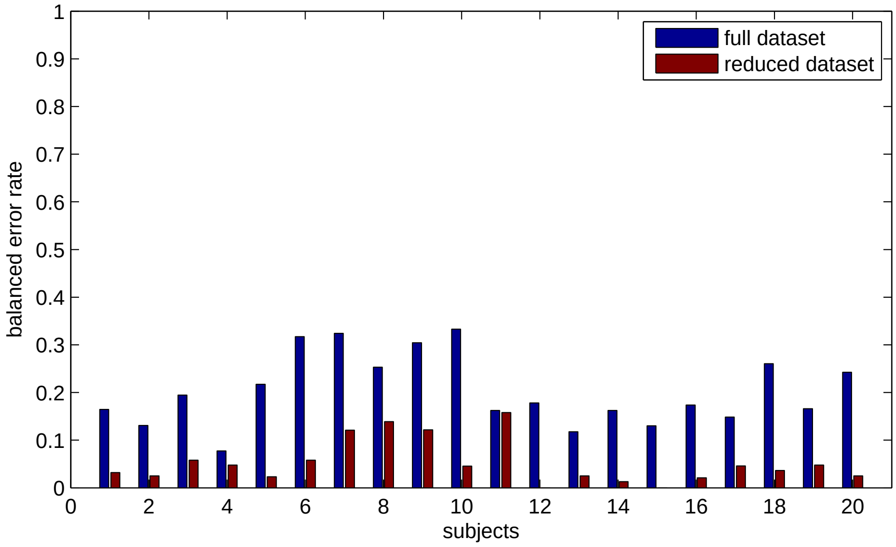

Figure 4.7: Results 2-class prediction with Active Learning using synthesis parameters

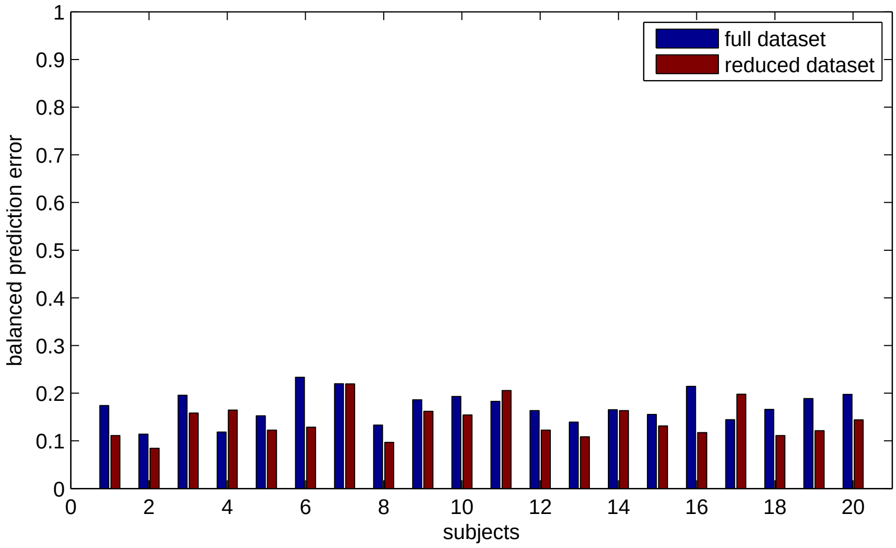

Figure 4.8: Results 4-class prediction with Active Learning using spike representation

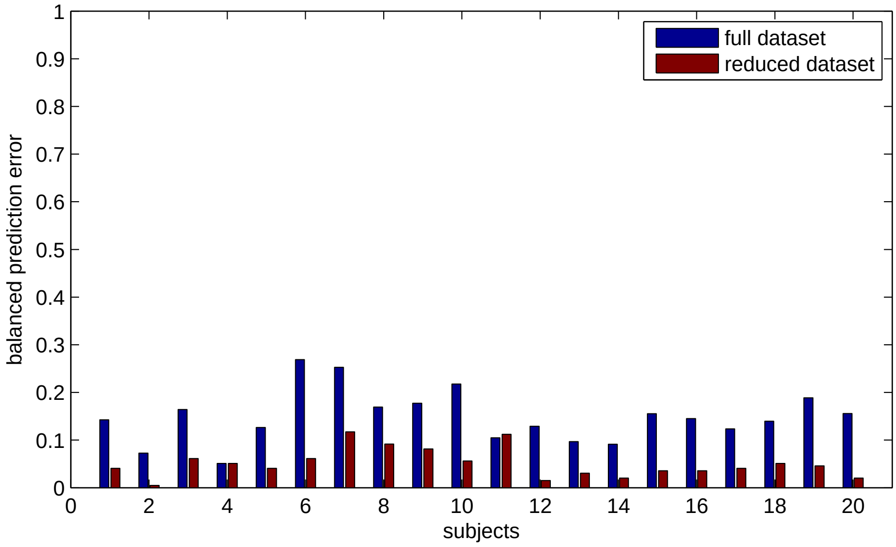

Figure 4.9: Results 2-class prediction with Active Learning using spike representation

5 Alternative Approaches

5.1 Adaptive Bottle Experiment

5.1.1 Introduction

Previous research on product-sound quality evaluation [4] shows that sound quality can be described as the *adequacy of a sound attached to the product.* This measure is the combinatory perception different qualities. As in the Bauhaus' notion, that basic design elements are always linked to their dynamics and only perceived together (line is a cause of a moving point [23]), we follow this principle in product design looking at sound, object and interaction, which are examined by their combined aesthetics. We investigate this by using prototypical design elements: The sounds generated are cartoonified, the object is an abstract bottle like vessel and the interaction is exemplified by a tilting gesture. There is a threefold interrelationship between these aspects: The bottle shape of the object induces a gesture which causes sound that gives feedback and influences the gesture. The changing sound in turn causes a different perception of the object itself (bottle empty/full) that affects the gesture (figure 5.1).

Sound design as the synthetic generation of sound and aesthetic decision making by controlling the synthesis parameters can be combined with the product and its usage by preference learning in a parameter mapping task [20]. In this respect, physically based sound design offers a novel alternative to recording sounds [40]. We measure adequacy of the generated sounds through judgments of subjects interacting with the object, and optimize the quality of the sounds using statistical methods iteratively.

Instead of an unguided trial and error process to find the best parameter settings in terms of adequacy to the product function, we propose to substitute it by a guided one, depending on preference judgments. Here, we show an abstract implementation of this idea: the Adaptive Bottle.

Adaptive Bottle is an interactive object, that has been designed to simulate the – suggested by the shape of the object – action of pouring [12]. It is connected to a physically based sound model that simulates the sound of drops or small objects hitting the surface of a resting liquid (dripping / splashing) [48]. The bubbles sounds support the interaction with the bottle. Therefore they are supposed to give the user the right feedback to measure how full it is.

The sound design task is to tune a physically based sound model to produce a desired sound by adjusting it's input parameters based on judgments.

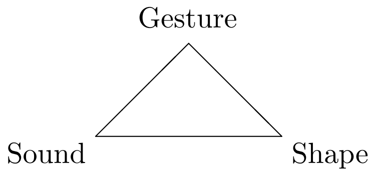

Figure 5.1: threefold aesthetics

5.1.2 Adaptive Bottle Optimization

The Adaptive Bottle has a built-in accelerometer, and communicates with the bubbles sound model via the wireless interface of the chip. The acoustic model in [2], based on the use of large quantities of bubbles to represent complex liquid sounds, has been implemented in the Max/MSP environment. It has seven input parameters, which control the statistics of bubbles size, intensity of emission, and rate of bubbles formation. These parameters are used for determining the characteristics of the sound. In this paper, the bubbles size and formation rate parameters are selected for the optimization. The other parameters have been kept constant.

The accelerometer sends 3D orientation information to the computer. By using this information, the tilting angle is calculated. Based on the tilting angle, the volume of liquid remaining in the bottle is calculated. Remaining liquid is used in turn to determine the current bubbles size and the current formation rate. By using the acceleration information, calculated formation rate and bubbles size, and the other synthesis parameters, which are constant, the physically based sound model generates the bubbles sounds. Figure 5.2 shows this information flow during the whole pouring action. At the beginning of the interaction, it is assumed that the bottle is full. As the subject tilts the bottle, liquid is poured out, and the bottle becomes slowly empty, depending on the tilting angle. The amount of liquid in the bottle and the emerging bubbles sounds are updated depending on the tilting angle. Intuitively speaking, the size of the bubbles emerging decreases, as the bottle gets empty during the action of pouring. The sound of larger bubbles in pitch is lower than of smaller bubbles. Therefore the bubbles size decreases, as the bottle gets empty.

Least Squares Optimization

The basic idea behind this optimization is to find the direction and the amount of the optimization step to be made in that direction depending on the evaluation of four sample points, which is supposed to improve the quality of the produced sound. We minimize the unknown preference function by gradient descent. Thereto we model the near surround of the sample point linearly. The direction of a learning step is the gradient of the linear least squares solution. The mathematical formulation of this method according to the Adaptive Bottle problem is as follows:

1. The four data points x_t^i around the central data point x_t obtained in the last iteration, which are to be evaluated by the subject are generated. Each of these

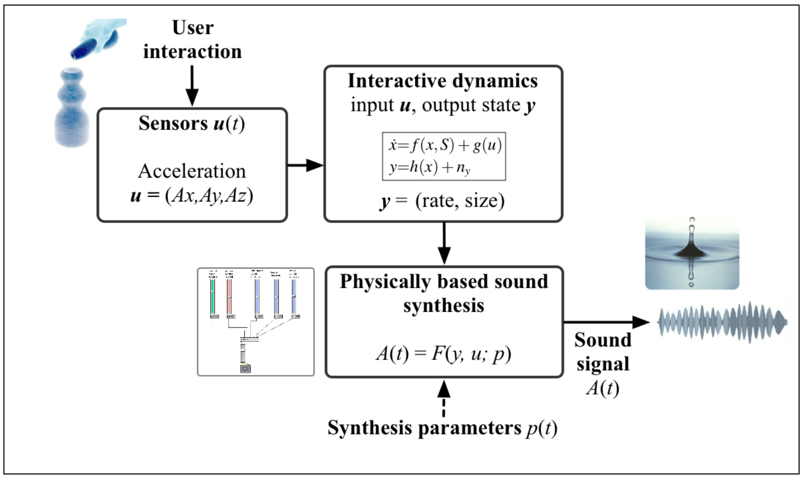
User interaction
Sensors $\boldsymbol{u}(t)$
Acceleration
$\boldsymbol{u} = (Ax, Ay, Az)$
Interactive dynamics
input $\boldsymbol{u}$, output state $\boldsymbol{y}$
$\dot{x}=f(x,S)+g(u)$
$y=h(x)+n_y$
$\boldsymbol{y}$ = (rate, size)
Physically based sound synthesis
$A(t) = F(y, u; p)$
Sound signal
$A(t)$
Synthesis parameters $p(t)$

Figure 5.2: The interaction between the subject and the Adaptive Bottle is shown. The acceleration parameters are used for calculating the new volume of the liquid remaining in the bottle, the formation rate and the size of the bubbles. The formation rate and the size of the bubbles are combined with the other parameters to generate the sounds.

four points are shifted to the origin, and each dimension is normalized separately.

$$\begin{aligned} x_t^i &= x_t + r_t \cdot \begin{bmatrix} \cos\frac{\pi \cdot i}{2} \\ \sin\frac{\pi \cdot i}{2} \end{bmatrix}, i = 1, \ldots, 4 && (5.1) \\ A &= \begin{bmatrix} x_t^{1T} \\ x_t^{2T} \\ x_t^{3T} \\ x_t^{4T} \end{bmatrix}, && (5.2) \\ Aw &= b, && (5.3) \end{aligned}$$

where w is the weight vector, b is the evaluation vector, and r is the radius. Using the evaluation values b, the optimal w values are looked for.

2. Squared error is calculated.

$$E = ||Aw - b||^2 \qquad (5.4)$$

3. The squared error calculated in Equation 5.4 is minimized, which yields the direction of the gradient in w.

$$0 = \frac{d}{dw}[||Aw - b||^2]. \qquad (5.5)$$

Taking the derivative, and solving the equation for w yields:

$$w = (A^T A)^{-1} A^T b. \tag{5.6}$$

4. The new central data point x_{t+1} is calculated. The learning step λ and the radius r are updated.

$$\begin{aligned} x_{t+1} &= x_t + \lambda w, && (5.7)\\ \lambda_{t+1} &= \lambda_t \cdot 0.9, && (5.8)\\ r_{t+1} &= r_t \cdot 0.9. && (5.9) \end{aligned}$$

The decrease of the learning step makes the move in the learnt direction shorter after each iteration. Decreasing the radius causes that the four sample points come closer to each other in the next iteration. Both of these decisions increase the accuracy of the learning. After a certain number of learning steps, the sample points are so close to each other that the difference between the emerging sounds for these samples is not audible anymore. The subject stops the experiment at this point.

Adaptive Bottle Experiment

The preference learning experiments have been performed on the subjects to test the applicability of such statistical methods for these kinds of optimization tasks. Each subject performed the same experiment three times with three different, preselected initial parameter settings. At the beginning of the experiment, four sample points are presented to the subject around the initial point. One sample point is on the left hand side, one on the right; the other two points are one up and one down (See Figure 5.3). The subject evaluates all of them one by one in a random order. The evaluation is supposed to be made in a comparative manner, since after the evaluation, the direction of the learning step is going to be chosen depending on the evaluation, i.e. the combined direction with the highest evaluation rates is calculated. The judgments have values within the interval $[0, 1]$.

The evaluation process is as follows: 4 parameter settings are available for judgment (see equation 5.2). The subject chooses and listens to them sequentially in an arbitrary order with possible repetition, while performing the action. The subject is able to set the judgment values of all 4 settings at any time, until he is satisfied with the preference ranking. After confirmation, the system will advance to the next 4 setting examples, which have to be judged again the same way. The flow of the experiment is shown in Figure 5.4.

The user can repeat this evaluation process arbitrarily many times until he / she is satisfied with the quality of the sound. In a typical session five or six learning steps are sufficient. When the subject decides to stop, the trajectory of the whole learning process is shown on a 2D parameter space diagram. The sound corresponding to initial parameter values and the final sound are played as well to show the improvement.

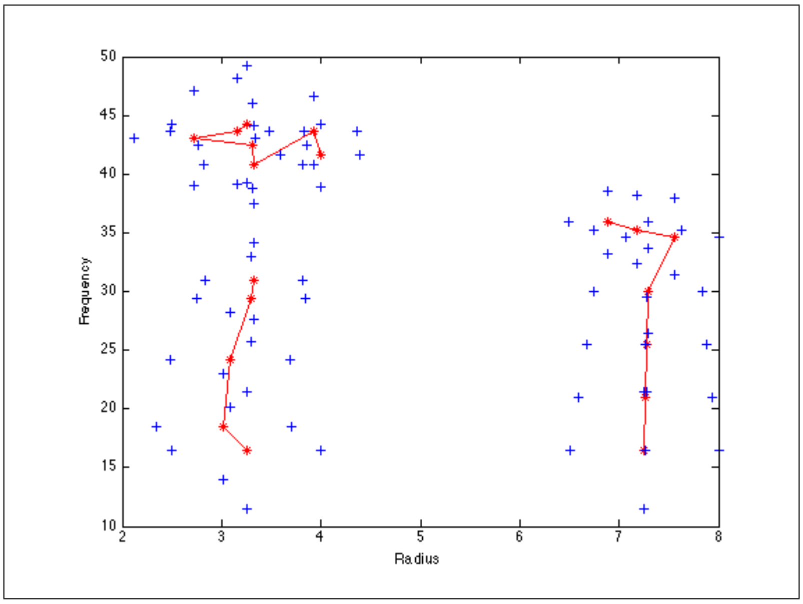

Figure 5.3: The results of one Adaptive Bottle experiment are shown. The curves and the dots on them are the learning curve (learning steps) and the data points. The plus signs are the four points calculated around the current data point.

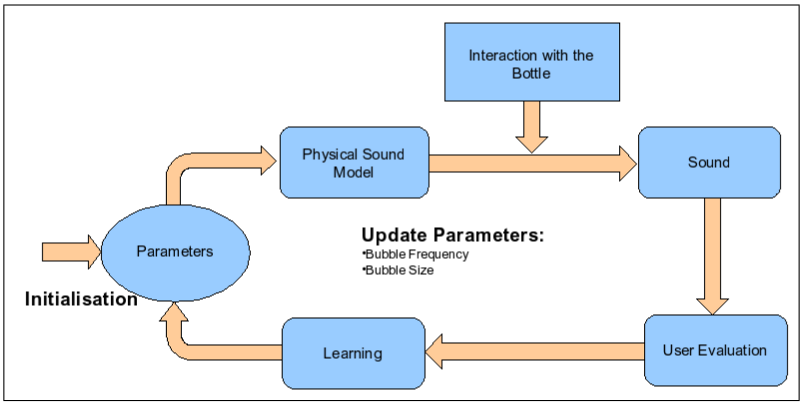

Figure 5.4: The complete scenario of the Adaptive Bottle experiment is shown.

Experimental Results

The preselected initial points given to the subjects were chosen to be as 1. small formation rate, small bubbles size, 2. large formation rate, small bubbles size, 3. small formation rate, large bubbles size.

The experiments were performed by 15 subjects in total. The summarized results shown in Figure 5.5 depict only the first and last points of each experiment, in order to show the tendency of each subject. This plot shows all of the three experiments

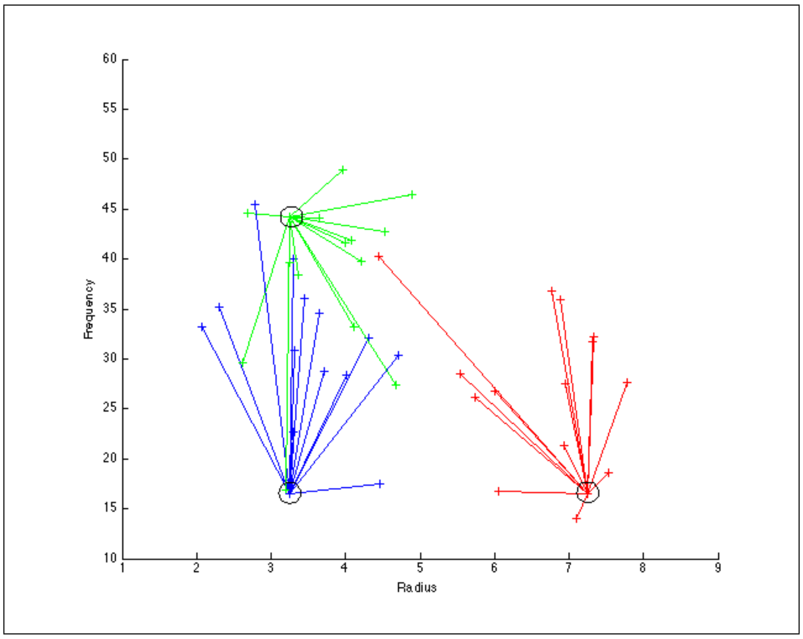

Figure 5.5: The summarized results are shown. The circles indicate the initial points.

performed by each subject. The bottom right lines depict the results of the experiment with large bubbles size and low formation rate. The bottom left lines are the results of the experiment starting with small bubbles size and small formation rate. Finally the top left lines show the results of the experiment with large bubbles size and small formation rate. The mean of the results for each experiments is shown in Figure 5.6. In this figure, the mean values of the end points of each subject are depicted for each experiment separately. The starting points are the same as in Figure 5.5.

As it can also be seen on Figure 5.5 and 5.6, the two plots with the small formation rate tend to move in the direction to increase the formation rate. One can also see that the change in the bubbles size increasing for the small bubbles size case, and decreasing for the large bubbles size case, however the main action happens in the vertical direction. The tendency of the curves to move in the vertical direction shows that the formation rate plays a more important role in these experiments than the bubbles size. As a consequence of that, three experiments performed by each subject do not define a closed 2D region in the parameter domain, but rather converge to a certain formation rate region, where the formation rate of the bubbles sound more realistic. For the cases, where the formation rate value is small, all subjects made moves in the direction of increasing the formation rate, whereas the bubbles size parameter was changed only a small amount compared to the change in the formation rate. However, for the case, where the formation rate is already large, the learning curves generally do not have a common direction.

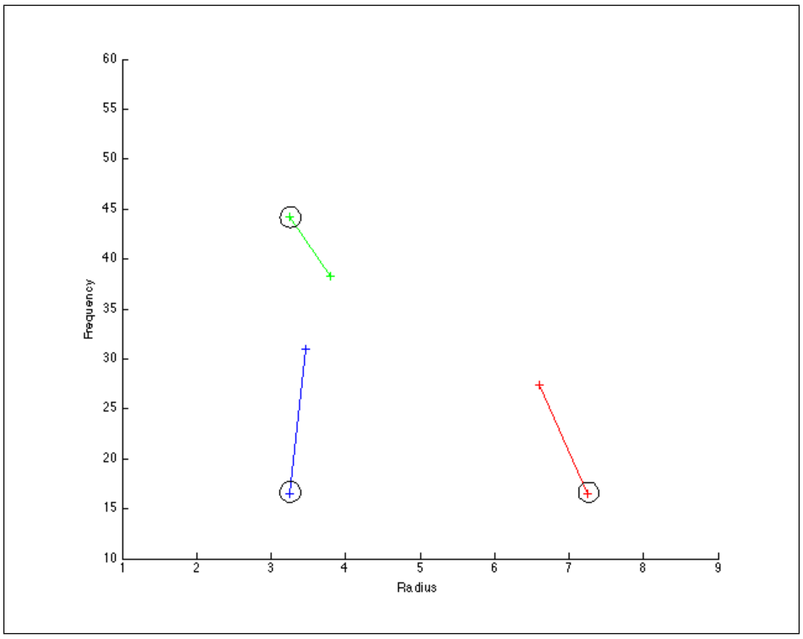

Figure 5.6: The mean of the results are shown. The circles indicate the initial points.

6 Conclusion & Discussion

6.1 Representation

We have investigated the potential of two gamma-tone based representations. The first representation is a structure preserving method, where we used graph theory to compare the representations of two sounds. The second representation scheme we proposed is a standard feature vector based representation. Furthermore two classification methods have been tested, which have been used in combination with these representations for classifying impact like everyday sounds.

The gamma-tone Meddis representation is a generic representation which in general gives good results. However, if a specific problem is defined, specialized descriptors, using psychoacoustic features outperform the gamma-tone Meddis representation.

From the theoretical view point, support vector machines try to optimize the classification boundary between two classes, by maximizing the margin between the two classes. In order to do this, the method tries to find the optimal input vectors, called support vectors, which maximize the margin. On the other hand, using a multi-variate Gaussian for approximation of the observation density, the hidden Markov models try to predict the state transition probabilities and the mean-variance pair for each dimension in the input parameter space. Hence, the search space of the hidden Markov models is much bigger than the support vector machines. Therefore the hidden Markov models need much more input samples to improve the prediction quality. Considering this fact, the results, which we obtained with the hidden Markov models are satisfactory.

The comparison of these experiments showed that, in the case of the step sounds, gamma-tone based representation methods performed well for the classification of the sounds no matter which classification method has been used. However, support vector machines performed in general better than hidden Markov models.

The sound corpus tested in these experiments is limited to only impact like footstep sounds. This corpus is not sufficient to draw general conclusions about the applicability of these representations on different categories of everyday sounds. Therefore further experiments are needed. A general everyday sounds database is going to be generated, which the representation methods can be tested on. Especially the spike representation is going to be tested on this general sound corpus in comparison with the state of the art representation schemes like the MFCCs or standard psychoacoustical descriptors etc.

The spike representation is a new approach to represent sounds. The most important step of the representation, namely the gammatone auditory filters are biologically motivated, since they simulate the basilar membrane motion. However, the matching pursuit algorithm, which has been incorporated to find the optimal positions of the spikes does not have a biological background. Furthermore, the stability of the representation against periodic continuous sounds is questionable. Therefore the representation will be extended into a more stable representation. In order to understand the perceptually relevant features in a better way, the representation will be adapted closer to an auditory image representation [38, 37, 36].

6.2 Adaptive Optimization

We investigated the potential of parameter optimization of a physically based model in product sound design. Based on the notion that the product quality can only be measured when sound, shape and gesture are examined together, we implemented an experimental setup. A local gradient based method on subjective judgments shows common effects over the subjects: The subjective quality is increased step by step and a principal direction in parameter space could be identified. Although the used optimization using a simple update rule, the results encourage to advance to a comprehensive psycho-acoustic evaluation of the matter.

Statistical methods provide more structure to parameter search problems. However in a 2D domain, random search can converge faster than such an algorithm. Besides, in a 2D domain, in order to make one learning step, four data samples are used. In a higher dimensional domain, this amount increases exponentially, when using two points for every dimension, which makes the problem intractable.

Therefore the model will be improved so that less number of data samples will be needed to make one learning step. Two data samples from the previous evaluation step can be used, and only two new data samples can be presented to the subject. However even this would not solve the problem in a high-dimensional case.

In order to solve the high-dimensional problem, the direction of the learning step should be estimated without evaluating all the data samples around a certain data point. The evaluations in the previous iterations can be taken into account, while presenting to the subject new data samples. Bayesian inference can be built into the model to use the prior knowledge obtained in the previous iterations to calculate a posterior probability of the direction of the next learning step. Hence, a more sophisticated, probabilistic machine learning model will be incorporated to extend the optimization to higher dimensional scale.

Acknowledgements

We thank the masters students Elena Martinez Hernandez, Vicky Ludlow and Elio Wahlen for their contributions to this deliverable.

Bibliography

[1] J. M. Adrien. *Representaion of Musical Singnals*, chapter The missing link: Modal synthesis, pages 269–297. MIT Press, Cambridge, MA, 1991.

[2] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966. first HMM article.

[3] B. Berlin and P. Kay. *Basic Color Terms: Their Universality and Evolution,*. University of California Press, 1969.

[4] J. Blauert and U. Jekosch. Sound-quality evaluation — a multi-layered problem. *Acustica – Acta Acustica*, 83-5:747–753, 1997.

[5] B. E. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992. The basis paper for SVM with kernel trick.

[6] J. Breebaart and M. McKinney. Features for audio classification. In *Proceedings of the Philips Symposium of Intelligent Algorithms*, Eindoven, 2002.

[7] J. Breebaart and M. McKinney. Features for audio and music classification. In *Proceedings of the International Conference on Music Information Retrieval*, Baltimore, 2003.

[8] P. A. Cabe and J. B. Pittenger. Human sensitivity to acoustic information from vessel filling. *Journal of experimental psychology: human perception and performance*, 26-1:313–324, 2000.

[9] L. H. Carney and C. T. Yin. Temporal coding of resonances by low-frequency auditory nerve fibers: Single fibre responses and a population model. *J. Neurophysiology*, 60:1653–1677, 1988.

[10] J. B. Carrol, editor. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*. MIT Press, cambridge, MA, 1956.

[11] E. de Boer and H. R. de Jongh. On cochlear encoding: Potentialities and limitations of the reverse correlation technique. *Journal of Acoustics Society of America*, 63:115–135, 1978.

[12] K. Franinovic and Y. Visell. Strategies for sonic interaction design: From context to basic design. In *ICAD '08: Proceedings of the International Conference on Auditory Display*, Paris, France, 2008.

[13] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[14] W. W. Gaver. What do we hear in the real world? *Ecological Psychology*, 5-4:285–313, 1993.

[15] B. R. Glasberg and B. C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47:103–138, 1990.

[16] R. L. Goldstone, Yvonne Lippaa, and Richard M. Shiffrina. Altering object representations through category learning. *Cognition*, 78-1:27–34, 2001.

[17] S. Harnad. *To Cognize is to Categorize: Cognition is Categorization.* Elsevier, 2005.

[18] R. Herbrich, T. Graepel, C. Campbell, and C. K. I. Williams. Bayes point machines. *Journal of Machine Learning Research*, 1(4):245–278, 2001.

[19] O. Houix, N. Misdariis G. Lemaitre, P. Susini, K. Franinovic, D. Hug, J. Otten, J. Scott, , Y. Visell, D. Devallez, F. Fontana, S. Papetti, P. Polotti, and D. Rocchesso. Closing the loop of sound evaluation and design (CLOSED): Deliverable 4.1, everyday sound classification: Sound perception, interaction and synthesis. Technical report, IRCAM (Paris), UNIVERONA (Verona), HGKZ (Zurich), 2007.

[20] H. Hunt, M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *NIME '02: Proceedings of the Conference on New Interfaces for Musical Expression*, Dublin, Ireland, 2002.

[21] K. H. Hunt and F. R. E. Crossley. Coefficient of restitution interpreted as damping in vibroimapct. *AMSE J. Applied Mechanics*, pages 440–445, 1975.

[22] Sound ideas sound database, http://www.sound-ideas.com.

[23] W. Kandinsky. *Point and Line to Plane.* Courier Dover Publications, New York, USA, 1979.

[24] D. H. Lawrence. Acquired distinctiveness of cues: II. selective association in a constant stimulus situation. *Journal of Experimental Psychology*, 40-2:175–188, 1950.

[25] G. Lemaitre, P. Susini, S. Winsberg, B. Leinturier, and S. McAdams. The sound quality of car horns: a psychoacoustical study of timbre. *Acta Acustica*, 93:457–468, 2007.

[26] M. A. Liberman and I. G. Mattingly. The motor theory of speech perception revised. *Cognition*, 21:1–36, 1985.

[27] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, 2000.

[28] Vicky Ludlow. Perceptive approach for sound synthesis by physical mdoelling. Master's thesis, Chalmers University of Technology, Gteborg, Sweden, 2008.

[29] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[30] E. Martinez, K. Adiloglu, R. Annies, H. Purwins, and K. Obermayer. Classification of everyday sounds using perceptual representation. In *Proceedings of the Conference on Interaction with Sound*, volume II, pages 90–95. Fraunhofer Institute for Digital Media Techology IDMT, 2007.

[31] R. Meddis. Simulation of mechanical to neural transduction in the auditory receptor. *Journal of the Acoustical Society of America*, 79-3:702–711, 1986.

[32] R. Meddis. Simulation of auditory-neural transduction: Further studies. *Journal of the Acoustical Society of America*, 83-3:1056–1063, 1988.

[33] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 1909.

[34] S. Delle Monache, D. Devallez C. Drioli, F. Fontana, S. Papetti, P. Polotti, and D. Rocchesso. Closing the loop of sound evaluation and design (CLOSED): Deliverable 2.1, algorithms for ecologically-founded sound synthesis: Library and documentation. Technical report, UNIVERONA (Verona), 2007.

[35] R. E. Pastore. *Categorical Perception: Some Psychophysical Models.* Cambridge University Press, New York, 1987.

[36] K. Robenson R. D. Patterson and J. Holdsworth. Complex sounds and auditory images. *In Proceesings of Auditory Physiology and Perception*, 9:429–446, 2004.

[37] R. Patterson. Auditory images: How complex sounds are represented in the auditory system. *Journal of Acoustics Society of America*, 21-4:183–189, 2000.

[38] R. D. Patterson and J. Holdsworth. A functional model of neural activity patterns and auditory images. *Advances in Speech, Hearing and Language Processing*, 3:547–563, 1996.

[39] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, Analysis/Syntesis Team, 2004.

[40] D. Rocchesso and F. Fontana, editors. *The Sounding Object,.* Mondo Estremo, Firenze, Italy, 2003.

[41] F. Rosenblatt. The perceptron: A probabilisitic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[42] B. Schulte-Fortkamp and D. Dubois. Recent advances in soundscape research. *Acta acustica united with acustica*, 92-6, 2006.

[43] M. Slaney. *An Eifficient Implementation of the Patterson-Holdsworth Auditory Filter Bank.* Apple Computer, 1993.

[44] M. Slaney. *A matlab toolbox for auditory modeling work.* Interval Research Corporation, 1998.

[45] E. Smith and M. S. Lewicki. Efficient coding of time-relative structure using spikes. *Neural Computation*, 17:19–45, 2005.

[46] C. T. Snowdon. *Categorical perception: The groundwork of Cognition*, chapter A Naturalistic View of Categorical Perception. Cambridge Univerity Press, New York, 1987.

[47] C. Spevak and R. Polfreman. Analysing auditory representations for sound classification with self-organizing neural networks. In *Proceedings of the International Conference on Digital Audio Effects*, 2000.

[48] K. van den Doel. Physically-based models for liquid sounds. *ACM Transactions on Applied Perception*, 2-4:534–546, 2005.

[49] N. J. Vanderveer. *Ecological Acoustics: human perception of environmental sounds.* Phd thesis, Cornell University, 1979.

[50] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[51] Y. Visell, K. Franinović, and D. Hug. Closing the loop of sound evaluation and design (CLOSED): Deliverable 3.1, sound product design research: Case studies, participatory design, scenarios and product concepts. Technical report, ZHdK (Zurich), 2007.

[52] W. H. Warren and R. R. Verbrugge. Auditory perception of breaking and bouncing events:a case study in ecological acoustics. *Journal of experimental psychology: human perception and performance*, 10-5:704–712, 1984.

[53] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models.* Springer-Verlag, Berlin Heidelberg NewYork London, 22 edition, 1990.