FP6-NEST-PATH project no: 29085
Report Version: 1
Report Preparation Date: July 07, 2009
Classication: Pub.
Deliverable no. 5.2

Closing the Loop of Sound Evaluation and Design (CLOSED)

Deliverable 5.2

Visualisation and Measurement Assisted Design

Kamil Adiloğlu, Robert Anniés,
Hendrik Purwins, Klaus Obermayer
Neural Information Processing Group (NIPG)
School IV – Electrical engineering and Computer Science
Technische Universität Berlin
with contributions from
Sara de Bruijn, Carlo Drioli, Olivier Houix, Cyril Laurier, Antoine Minard,
Nicolas Misdariis, Matthias Schultze-Kraft, Yon Visell, Elio Wahlen

Contents

1 Introduction

If a sound makes you relax - when listening to the waves on the beach - or lets you shiver - like scratching fingernails on the blackboard - makes a crucial difference when creating the sounds for an interaction with a physical object, a video game, a movie, or a cyber instrument. Being a sound designer, you may have Gigabytes of sounds on your hard drive or a high number of parameters for your sound synthesis model to tune. But imagine a painter with hundreds oil paint tubes all piled up with no labels on them informing you about the color content. How tiresome if you have to open each tube until you find the appropriate color. Imagination would be much more inspired if the perceptual space of different functional-aesthetic, and emotional shades could be mapped intuitively on a two-dimensional plane, on a color palette for perceptual sound attributes. If you use sound synthesis models, be it spectral, granular, physical, general algorithmic models, ...: in the end there are parameters to set (mostly using some sort of sliding controller or just number inputs) to fine tune or even to make the model work inside the specifications of audible signals. At this point the complexity of the model, i.e. the number of parameters or its dimensionality determines the utility of it. Moreover, it can determine if a model is feasible at all for sound design.

How can a machine assist the sound designer with methods that represent audio signals visually and make them accessible to manipulations using visualization? We have shed light to the topic from different perspectives.

Sound Representation We present two representation types for everyday sounds: a sparse sound decomposition into sound atoms (Section 2.2) using a biologically plausible filter database (The spike code representation, introduced in Deliverable 5.1 [2].) and a large set of psychoacoustically inspired descriptors (Section 2.3).

Sound Visualization and Navigation in the Soundscape This sound decomposition technique is considered as a tool to reveal the morphology of a sound as a sort of skeleton or cartoon. This skeleton shows the temporal pattern of a sound in an easy to understand way. In [17], we demonstrate the potential of this representation technique for navigating in the sonic space of the physical synthesis model. Another possibility to navigate in the sonic space is to reduce the dimensionality of the parameter space using discriminant analysis methods. The reduced parameter space can be visualized on a two-dimensional plane (See Section 4.2), where the user can navigate easily.

Pairwise Clustering of Spike Coded Sounds Following the objective to develop intuitive visualizations, we investigated the extraction of representatives of sound categories.

With this respect, it is possible either to find the most central example in a particular sound category or to generate an artificial example as a prototype or representative for that category. This representative can serve as an icon for this class if it can be perceived as such. If the data (sound events) is available as (feature) vectors in Euclidean space (See Chapter 4.2) we can apply linear projection methods. In the more general case however, only pairwise dissimilarities can be assumed that do not necessarily fulfill the requirements for a metric. This situation occurs for example if subjects are asked to choose between two sounds, which they like better. The spike code representation and the corresponding distance measure lacks the properties of a metric. Hence, we cannot simply calculate distances in the Euclidean space and therefore we applied pairwise clustering. Our spike code representation enables us to perform clustering on a higher level than directly on the audio signal and therefore clustering can recognize structures (categories) there. We performed pairwise clustering on everyday sound categories to find out what sub-categories there are. The spike code representation in conjunction with the pairwise clustering method enabled us to generate these so-called sound icons for each detected sub-category. A sound icon depicts characteristics of a particular sound category, hence is able to represent it in comparison with the other sound categories.

Automatic Selection of Perceptual Dimensions In order to generate meaningful sound display we have applied feature selection to psychoacoustically inspired timbre descriptors. We use the classification performance as a hint which feature selection method to use. The superior performance of the Support Vector Machine (SVM) leads us to use SVM-based Recursive Feature Elimination [33] as a feature selection tool in Section 2.3.

Measurement Assisted Design In a closed-loop sound design, our machine should guide the user to the desired sound without time loss. The preference of the user - of whatever characteristics - should be inquired by the system in a minimized number of crucial queries addressed to the user, like in the children's game where one has to guess an item with as few yes/no questions as possible. Techniqually, we set out for this objective, employing active learning and online optimization techniques (Chapter 5). A stochastic regression method is made use of in this chapter to guide a user to navigate within the parameter space in a more efficient way. By doing this, the user can easily find the correct parameter values for the optimal sound.

Stimuli We have used the approaches explained above to investigate two sets of perceptually relevant categories, namely functional-aesthetic and emotional attributes. For the functional-aesthetic categories we chose the taxonomy based on the aggregate states of the materials and the type of interaction involved in the sound generation [26]. We use a 10-class taxonomy based on manually labeled data (Section 2.2) and a 5-class taxonomy validated in a psychological forced-choice experiment (Section 2.3). For the emotional categories we used the manually annotated categories popular in research of emotions in music (*happy, sad, aggressive, relaxed*). Both perceptual and cognitive categories are highly influenced by semantics, e.g. the recognition of a particular object through sound

or the recall of an emotional memory. We restrict ourselves to the investigation of purely acoustic features, since the consideration of models of context and semantics are beyond the CLOSED project.

Summary and Organization of Document In the sequel we have organized some of the main reports and publications generated during the second reporting period of the CLOSED project. We present predictors and visualization for functional-aesthetic attributes (Section 2.1), including the compilation of a general everyday sound database for evaluation. We will explain how we train classifiers with different representations, especially Spike Code Matching, MFCC, and timbre features. In Section 2.3, we evaluate dimension reduction via feature selection. Predictors and visualization for emotional attributes are presented in Section 3. An emotionally labeled music database is used to train a classifier using timbre features. Then a feature reduction method is applied to extract features that have most impact on the class decision. A two-dimensional visualization is provided by Isomap. In Section 4 about visualization techniques, we investigate how everyday sound database prototypes for the base classes are extracted, by pairwise clustering based on the Spike Code Matching representation. Fisher's Linear Discriminant (LDA, Section 4.2) is applied to impact sounds of the SDT impact model to construct a 2-D controller (Max/MSP) for the sound model that controls its 10-dimensional input. For measurement assisted design (Section 5), we introduce a demonstrator for Active Learning using the SDT rolling model. Least squares as an optimization technique for the Adaptive Bottle are evaluated in Section 5.3. A global optimization technique via Gaussian processes are examined with respect to the problem of parameter space search in Section 5.4.

2 Predictors and Visualization for Functional-Aesthetic Attributes

2.1 Introduction

When listening to sounds, humans can focus on the qualities of acoustical properties of sounds or identify the sound events and/or their properties. Considering these different ways of perceiving sounds, Gaver [27] introduced the distinction between musical listening and everyday listening. The research on everyday listening had appeared since answering questions regarding recognition and perception of everyday sounds was difficult by focusing solely on musical listening or speech recognition. In her pioneering work, Vanderveer [75] defined everyday sounds as *any possible audible acoustic event which is caused by motions in the ordinary human environment.* Vanderveer studied the perception of environmental sounds in an identification experiment. She played recorded sounds to subjects and let them describe what they had heard. She observed that the subjects mostly referred to the event that caused the sound, i.e. the action, the objects involved, and the place of the action [40]. Houix et al. [39] and Lemaitre et al. [47] performed a free categorization experiment in which subjects were asked to categorize kitchen sounds. The subjects also gave account whether they had categorized sounds according to the acoustic signal properties of the sound or to the event. They found out that naive listeners tend to group sounds according to the everyday listening mode. Consequently, all these psychoacoustical studies revealed that human beings identify everyday sounds in three different ways [37]:

- Acoustical similarity
- Event similarity
- Semantic similarity

With the acoustical similarity, the spectral similarity is meant. Event similarity describes that the events causing the emerging sounds are similar. Semantic similarity is associated to the objects depending on some knowledge about the sound causing event or meaning of that event. Hence, by grouping sounds together, spectral properties of the sounds have been used without or with very little interpretation. On the other hand, more interpretation of human beings are involved in event similarity as well as in semantic similarity. The identification of the sound causing event, in which situation or environment the sound emerged, who caused the sound to emerge are all important in

this interpretation. For this, an extensive context information is needed. This information is either supplied by the experimenter, in a psychoaouctical experiment, or human beings already own this knowledge mainly through experience.

Recently, the computational analysis of everyday sounds attracted increasing research interest [7]. There are fundamentally different approaches to the problem: namely spectrally based algorithms and models that consider an agglomeration of psychoacoustically inspired temporal and spectral descriptors, some of them taking into account basic operations such as (first and second) derivatives, mean, variance, skewness, and time series analysis methods. Furthermore, there are (biologically plausible) dictionary-based methods. However, one common aspect of all these studies is that they consider only the acoustical similarity due to the lack of context information, in which the analyzed sounds emerged.

Spectrally Based Descriptors Aucouturier et al. [4] used the bag-of-frames approach to model urban sound scapes. They approximate the distribution of the MFCC's over all frames by a multivariate Gaussian Mixture Model. Then they determined the similarity between two sounds by calculating the Kullback-Leibler divergence between their estimated densities. Since time-reversed sounds yield an identical representation according to this approach, it is appropriate for a sound ambience of static characteristics but does not account for temporal evolution.

Automatic Feature Selection and Generation Peeters and Rodet [57] suggest the use of discriminant analysis and mutual information for feature selection with subsequent maximum likelihood estimators for the classification of musical instruments. Defréville et al. [16] use genetic algorithms that combine basic operators and Mel Frequency Cepstrum Coefficients (MFCC's) to build optimal descriptors for the recognition of urban sound sources, using k-Nearest Neighbors and Gaussian Mixture Models as classifiers. However, the number of possible combinations of operators explodes. In our approach we will use feature selection implicitly and explicitly. In contrast to other classifiers, such as k-Nearest Neighbors, Support Vector Machines (SVM's) tolerate features irrelevant to the classification problem to a certain extent, without degrading classification performance. Recursive Feature Elimination [33] explicitly uses SVM for feature selection.

Sound Morphology Dufournet et al. [18] have outlined a procedure for sound source recognition, based on morphological signal descriptors. This approach has been refined by Peeters and Deruty [56], suggesting a straight-forward method employing dynamic, melodic and repetition profiles to describe the morphological structure of sounds (e.g. ascending, descending, impulsive, stable), using features such as loudness, pitch, lag-matrix-periodicity. In our approach, we are interested in a more generic method that uses specialized music descriptors such as pitch and loudness measures among a great variety of other ones. This makes the approach more flexible to be adapted to new problems.

Time Series Analysis Few authors have used other time series analysis for the representation and classification of everyday sounds. Hidden Markov Models in conjunction with Gaussian Mixture Models work well for speech recognition when very large data bases are available for training. Gaunard et al. [25] used Hidden Markov Models with LPC cepstrum and vector quantization to classify sounds of vehicles (car, truck, moped, aircraft, train). Anniés et. al. [3] employed Hidden Markov Models with a small number of hidden states to classify step sounds from different shoes on various soils. Hidden Markov Models, however, have a large number of parameters (transition probabilities, covariance matrix and mean of the Gaussian mixtures) to be estimated. Therefore this method does not perform well, if the data set is too small to reliably estimate a large set of model parameters. Spevak and Polfreman [67] equalized sounds in pitch, duration, and loudness. Frame-wise MFCC's are calculated and discretized by the Self-Organizing Feature Map, thereby translating sounds into symbol sequences of equal length. Two sounds are considered matching, if the corresponding symbol sequences can be transformed into each other by using at most k edit operations (k-difference inexact matching algorithm). A necessary assumption in this work is that all sound samples have the same length. But in practice, usually two sounds do not have equal duration and need to be aligned. Cowling and Sitte [13] compared various basic feature extraction methods, such as Short-Term Fourier Transform (STFT), Mel Frequency Cepstral Coefficients (MFCC's), Continuous Wavelet Transform (CWT) in conjunction with Dynamic Time Warping (DTW), Feed Forward Networks, Learning Vector Quantization, Gaussian Mixture Models (GMM's) or long term statistics (mean and covariance). Their evaluation on a small data set of footstep and other everyday sounds yields best results for MFCC's in conjunction with Dynamic Time Warping (DTW). Temko et al. [69] performed audio event classification for 16 meeting room sounds. They compared feature sets and then applied SVM's with various DTW kernels, the Fisher score kernel, and the Fisher ratio kernel. A Directed Acyclic Graph classification scheme is used as a multi-class scheme.

Sound Atom Dictionaries Several representation schemes have been proposed for representing different sound categories, in particular music, speech as well as everyday sounds. Sound signals have been represented sparsely in terms of a small number of atomic functions. Gribonval and Bacry [31] use harmonic atoms, which are adapted to encode musical instruments with harmonic overtone structure. But harmonic atoms are less usable for encoding environmental sounds. Chu et al. [9] proposed a sparse approach using Gabor atoms, applied for classification of environmental sounds via k-nearest neighbors and Gaussian mixture models. They apply matching pursuit to decompose the signal into n most prominent Gabor atoms. Mean and variance of frequency, scale, and translation position of these atoms are calculated, yielding a feature vector as a representation for classification. By representing the signal by means and variances of the parameters of its atomic decomposition, the detailed spectrotemporal structure of the signal gets lost. In contrast to Gammatone functions, Gabor atoms are symmetric in time. For sounds with fast attacks and slow decays (such as many impact sounds with reverberation), a decomposition with Gabor atoms introduces an artifact prior to

the attack of the signal, whereas the Gammatone function, itself with a faster attack than decay, constitutes a more suitable dictionary for this kind of sounds. Coifman and Wickerhauser [12] introduced Shannon entropy to select optimal basis functions out of a library of orthogonal wavelet-packets and localized trigonometric functions. In contrast to these approaches, we use biologically plausible basic functions, namely the Gammatone functions. We base our work on Smith and Lewicki [65] who introduced a sparse decomposition of audio signals into Gammatone components that resemble characteristics of cochlea filters.

Biologically Inspired Models Several other studies have been developed for explaining the auditory perception or hearing processes in a biologically plausible way [59]. Solbach et al. [66] have suggested a wavelet filter bank built from Gammatone filters. Meddis [52] developed a hair cell model, which has been adopted widely [51] for classification tasks of different types of sounds. Coath et al. [11] have suggested a cortical filter for onset detection. Only a few of these models have been applied onto a data bank of a reasonable number of audio samples, even fewer to everyday sounds [51].

We consider only single basic sound events omitting the problem of segmentation or separation when several sound events appear simultaneously or one after another.

2.2 A Graphical Representation and Dissimilarity Measure for Basic Everyday Sound Events

[1]

We focussed on representation, similarity, visualization and categorization of everyday sound events. We collected samples of everyday sound events to generate an everyday sound database according to the everyday listening phenomenon, conform to the studies of Gaver and IRCAM. In order to analyse these sounds in detail, we combined the visual and representational aspects of the sparse methods in a biologically motivated way. A Gammatone filterbank [54] is used as a basis for the sparse representation. The filter shape of a Gammatone filter mimics the time course of excitation on the basilar membrane, where the equivalent rectangular bandwidth (ERB) scale [28] of the center frequencies of the complete filterbank determines which region of the basilar membrane is excited.

The proposed coding scheme has been used for visualizing as well as for representing everyday sounds in a sparse way. The representation has been tested in a binary one-vs.-one as well as a one-vs.-all classification scenarios using a supervised machine learning algorithm. We developed a novel dissimilarity function for measuring the distance between two sparsely coded sounds without damaging the sparse, graphical structure of the code. Comparative simulations have been performed with standard spectral and

[1] *from:* Robert Anniés, Kamil Adiloğlu, Elio Wahlen, Hendrik Purwins, and Klaus Obermayer: A Graphical Representation and Dissimilarity Measure for Basic Everyday Sound Events, submitted to IEEE TASP

class	no.	subclass	no.	kind of event
solid	333	impact	92	single
		motor	116	continuous
		deformation	55	single
		friction	70	continuous
gas	175	wind	45	continuous
		whoosh	49	single
		explosion	81	single
liquid	169	drip	39	multiple/continuous
		flow	51	continuous
		pour	79	continuous

Table 2.1: The complete taxonomy everyday sounds and the number of sounds used in the evaluation of each class are shown.

psychoacoustical representation methods to assess the accuracy of the new method in comparison to standard state-of-the-art approaches.

For the details of this study, please refer to the Deliverable 5.1 [2] of the CLOSED project.

2.2.1 General Everyday Sounds Database

Different from the environmental sounds, which audibly describe certain environments like streets, coffee shops, highways etc., basic everyday sound events describe materials and their elementary interactions. Therefore, compared to the environmental sounds, everyday sound events are recordings of a single event and not a complete environment, which possibly contain many everyday sound events.

We exemplified IRCAM's everyday sounds taxonomy also considering Gaver's taxonomy proposal by collecting everyday sounds to generate an evaluation database. We selected the sounds from the Sound Ideas database. During selecting the sounds for a certain class, we have taken not only the acoustical similarity of the sounds into account, but also the event similarities. However semantic similarities have been discarded, in particular for sounds, which are acoustically different, eventhough they are semantically related, because we did not have either the necessary context information (Only the semantic descriptors of the database were available, which are insufficient for such purposes.) or the labels given to the sounds by human beings. Table 2.1 shows the classes and numbers of sounds per class.

The semantic descriptors of the Sound Ideas database are used for labeling the sounds. We used a hierarchical categorization with two levels. The first level of this hierarchy consists of the three main categories identified by Gaver: *solid*, *liquid*, *gas* and validated in an experiment by Houix (IRCAM) [40]. The aggregate state of the object or matter that is causing and supporting the emission of a sound is the discrimination criterion.

Solid

Sounds emitted by *solid* objects, like machines, engines and other mechanical devices are collected in this class. According to Gaver, the criterion for classifying a sound into this class is that the sound must be caused by mechanical contacts of one or more solid objects. The *solid*-class consists of four sub-categories. These are *impact*, *motor*, *deformation* and *friction*.

Impact-sounds are short sounds resulting from single instantaneous contacts between two objects or parts of one object. We selected recordings of switches, single typewriter clicks and hitting sounds. The *motor*-class consists of sounds of car engines in the idle state. Gaver indicates the regularity in the rolling sounds, which is the analogy to the *motor*-class. The fact that a combustion engine produces its sound mainly by a chain of explosions does not imply that we should treat it as an *explosion* sound. The perception of single explosions (see later) is completely different. Car crashes and glass crashes sounds constitute the *deformation* class, whereas the *friction* sounds category consists of recordings of squeaking and sliding doors, windows and dragging of objects. All such sounds involve an enforced contact of two objects moving against each other.

Since the motor sounds are continuous, we cut 4 sec. segments of these sounds. The friction sounds are selected to be shorter than 4 sec.

Liquid

Physically, liquids do not emit much sound themselves. Still there are a lot of everyday sounds that are typical for liquids and caused by water indirectly. In fact, the main sound sources are bubbles of air inside liquids that start oscillating. Alternatively, sounds emerge when liquids are reflected from or filled into solid objects making these objects oscillating. However, sounds of liquids are perceptually distinct.

The liquid sounds are split into three sub-classes: *drip*, *flow* and *pour*. The *drip*-class consists of recordings of dripping water. All examples have multiple countable drips (e.g. from a tap) or uncountable drips (e.g. rain). The recordings also differ mainly in the reverberation of the room and the material type (liquid itself or solid), where the drips are reflected or absorbed, i.e. at the point of sound emission. Hence the distinguishing factor of the *drip*-class to the other *liquid*-classes according to the semantics of the material interaction is: Small distinguishable portions of water fall onto a surface.

The *flow*-class consists of examples of running water taps, waves on the shore, rivers and similar movements of water. The class encapsulates movements of large portions of water that create swirls of air causing sound.

The third *liquid*-class is called *pour*. In there, we collected sound recordings of the interaction of transporting a portion of water from a vessel A to a vessel B through air, for instance filling a glas from a bottle. Sound is emitted by air bubbles that appear during the action. This definition seems close to that of the *flow*-class, but we have selected only mid-sized amounts of water/drinks (more than a drip, less than wave) and there must be a clear directed relocation of the water or drink from A to B. Because there are also sparkling drinks involved, the sounds are a mixture of the pouring action

and the sounds that comes from the carbonated drink itself.

We cut 4 sec. segments of these sounds, because liquid sounds are continuous.

Gas

Aerodynamic sounds sources are more direct. These sources create the sound by changing the atmospheric pressure. This can happen suddenly or as a steady process. The former are explosive sounds that populate the *explosion*-class. Gun shots and larger detonations from TNT are possible candidates for this sub-class proposed by Gaver. Explosions result in a very energetic short bang, that is almost a Dirac impulse. The strong and long reverb form detonations is the most typical feature. As *wind* we considered sounds resulting from a constant movement of air: recordings of wind sound at different locations, steam, and blow sounds. The third class is labeled as *whoosh*. It differs from the *wind*-class in its transient characteristics. The sounds of this class are caused by transient air blows (e.g. flame thrower) or objects moving fast through the air with relative velocity (e.g. arrow) to the listener (microphone) with respect to the sound source.

Similar to the motor and liquid sounds, we cut 4 sec. segments of the wind sounds.

2.2.2 Visualization of the Everyday Sounds Database

The sparse representation scheme, the so-called "*spike code*" we proposed in Deliverable 5.1 [2], has been used for visualizing the everyday sounds database described in the previous section.

Figure 2.1 shows spike codes of one sample sound from each class of the everyday sound database. Considering not single spikes but the spike code as a whole reveals that spike codes indicate common patterns for sounds with common attributes. Hence, spike codes of sounds having similar auditory features, e.g. being from the same class of a sound database, show similar patterns, whereas spike codes of sounds of different classes are significantly different. Hence, spike codes give a significant idea about the temporal pattern of a particular class of sounds. Departing from this fact, the spike codes can be regarded as a representation to perform classification of these sounds by making use of these similarities between them.

2.2.3 Automatic Classification of the Everyday Sounds Database

A sparse representation scheme has been applied for visualization of everyday sounds succesfully. Since the patterns of the sounds from each class depict similarities to their class members, these similarities can be made use of in classification scenarios to categorize them automatically.

In typical classification scenarios, given data is coded in vectors of certain features, and these feature vectors are classified in Euclidean space by using a diversity of machine learning algorithms. However, for sound classification problems, the spike code similarities offer a totally new paradigm, which enables to solve this problem by calculating these similarities without destroying the original structure of the pattern. Hence, a structure preserving distance measure is needed.

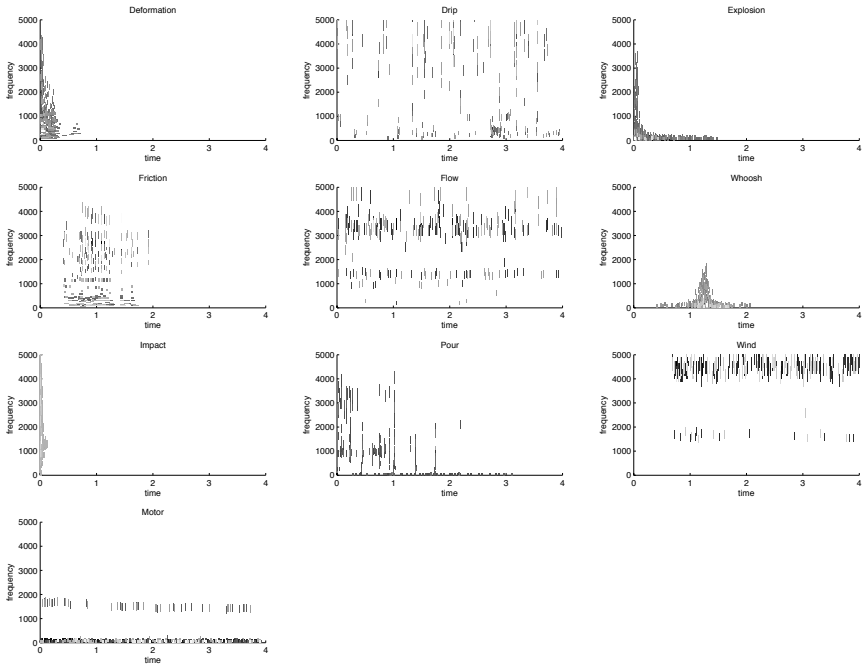


Figure 2.1: Spike codes of one sample sound from each class of the everyday sounds database are shown. These spike codes are generated by using a Gammatone filterbank of $M = 256$ filters. Each spike code contains $K = 256$ spikes.

For this purpose, we propose a novel, structure preserving distance measure between two spike codes for calculating the distance between the corresponding sounds. Having sounds encoded as spike codes, we define the distance in two steps. In the first step, the distance between two spikes of different spike codes are calculated (For the details refer to the Deliverable 5.1 [2]).

In the following step, the total distance between the given spike codes is calculated by taking the minimum of the sums of spike distances of all pairwise assignments. Intuitively, this method measures the minimal effort to transform one spike code into the other in terms of the single spike distance. Instead of calculating this minimal effort directly, we consider the spike codes of two given sounds as two point graphs combined in a bipartite graph. In this constellation, the vertices are the spikes of two spike-coded sounds, where each sound corresponds to a disjoint subgraph of the bipartite graph. The weights (similarities) between them are derived from the single-spike dissimilarity calculated in the first step. This consideration converts the problem into a combinatorial one of finding a perfect matching of the weights in a bipartite graph by the Hungarian algorithm [44]. We consider a matching to be a subset of the edges of the given graph containing each vertex only once. In a perfect matching, every vertex within the graph

is adjacent to an edge. The Kuhn-Munkres Theorem [44] guarantees the convergence of the algorithm to a perfect matching.

Both the spike coding algorithm and the Hungarian algorithm are time consuming operations. Coding the sounds with too many spikes takes a large amount of time, which in return does not necessarily yield either a better visualisation or a better classification accuracy. Therefore, it is essential to determine the optimal number of spikes. For this, we performed experiments with different number of spikes. For this we chose the *drip* and *flow* classes. We coded the sounds within these two classes with 32, 64, 96 spikes up to 256 spikes and calculated the balanced test accuracies of classification experiments performed with these codings. The results shown in Figure 2.2 indicate that the accuracy increases as the number of spikes increases. However there is a plateau between 160 and 224 spikes. For the sake of computational costs – the Hungarian algorithm has complexity of $O(K^3)$ – we chose to run the experiments with 192 spikes. This number had an acceptable accuracy as well as computation time.

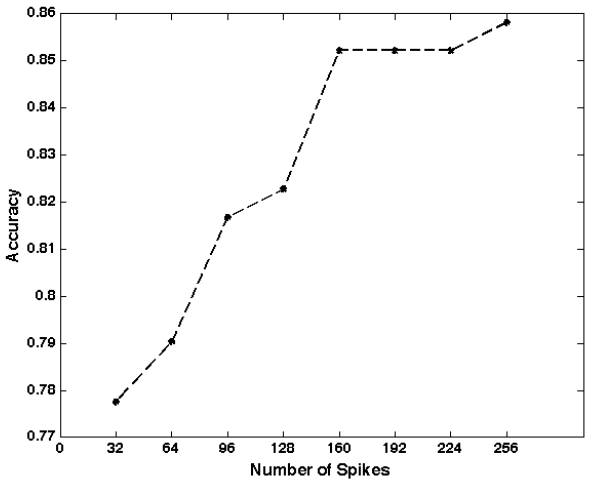


Figure 2.2: The balanced test accuracies for the classification experiments of the sound classes *drip* and *flow* are shown for different number of spikes used for coding the sounds.

To overcome the disadvantages of block based methods, we used a hop size of one sample during the coding to find the best fit and to be able to convolve the signal with the filter at once rather than to iterate over all hop positions. We found that using hop size values larger than 10 samples, the difference in the quality between the original and the resynthesized sound is clearly audible.

We employed this distance calculation method for all classification experiments. The sounds, however, could be better classified if the coefficients were tuned specifically to each class. E.g. *impact* and *whoosh* have very different lengths. Forcing the distance to measure length only would lead to better results in this task. Depending on the application this can be useful to do, here we wanted to measure the overall performance of the method without embedding too much prior knowledge of the classes during training.

		MFCC	SPIKE	SLL	TIMBRE
one vs. one	μ	97.70	94.86	92.44	80.29
	σ	2.4	5.5	5.7	9.4
one vs. rest	μ	89.19	87.51	81.32	55.68
	σ	7.8	5.2	10.71	7.4

Table 2.2: Mean and standard deviations of the results

In order to evaluate the accuracies of the dissimilarity matrices as well as of the three representations we mentioned in Section 2.3.2, we have performed classification experiments with the P-SVM to discriminate one class from another, for all pairs of classes separately, 45 pairs in total. Furthermore, we have performed detection experiments, again with P-SVM, where we measured the detection accuracies of one class vs. all other classes, for all classes separately, 10 experiments in total. The hyperparameter ϵ of the P-SVM and the kernel size γ were varied in a grid search: $0.1 \leq \epsilon \leq 1.0$, $2^{-5} \leq \gamma \leq 2^{10}$ to find an optimal setting.

We measured the accuracies of all these experiments by using the leave-one-out cross validation method.

2.2.4 Classification Results

Figure 2.3 shows the accuracies for binary discrimination tasks between all pairs of classes in an upper triangular form as well as the binary detection accuracies underneath as bar plots. Each bar corresponds to one representation method, namely to the MFCCs, spike representation, SLL and TIMBRE descriptors respectively. Table 2.2 shows the means and standard deviations of these results.

The results of the one vs. one discrimination tasks indicate that the overall performance of the MFCC representation outperforms the other three representations, including the spike representation. However, the overall accuracy obtained by the spike representation is very close to the MFCC accuracy. Furthermore, for some particular cases, for *friction* vs. *impact* for instance, the spike representation outperforms the MFCC representation. The one versus all others experiments show the detection performances of the classifiers. In the detection performances, again the MFCC representation outperforms the other representation schemes on average. However, the results of the spike representation are very close to the results obtained by the MFCC representation. Similar to the results of the discrimination experiments, for some cases like *friction, drip, flow, whoosh* and *wind*, the spike representation outperforms the other representations.

SLL and TIMBRE show mixed results. Only SLL has a fairly good performance in the one vs. one task. A high variance and/or poor accuracy of SLL and TIMBRE in the other cases make them less applicable here.

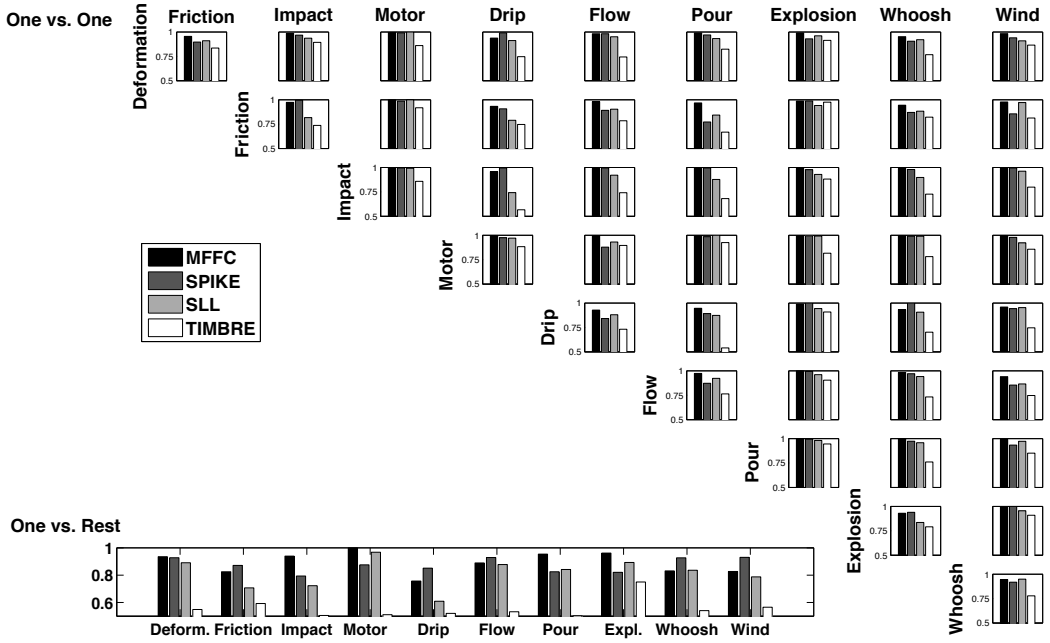


Figure 2.3: The upper figure shows the results of the one vs. one discrimination experiments, the lower figure shows one vs. rest results. The values are balanced prediction rates (1 - class balanced error). MFCCs and SPIKE perform generally best in the one vs. one task with MFCCs slightly better than SPIKE. The one vs rest task in 5 cases SPIKE performs best, MFCCs win the other 5.

2.2.5 Discussion

Noise The amount of noise in everyday sounds can't be neglected in the analysis and classification. How is noise encoded with the gammatone spike approach? Firstly, the encoding can be understood as a denoising step that emphasizes the contours within the sound which are perceptually important and result in a robust distance measure. Secondly, from a certain threshold of spike numbers the noise in the signal is encoded in a probabilistic way. The position of a Gammatone atom in the noise is determined by the noise distribution. Encoding sounds with an inherent portion of noise, like streaming water, could emphasize contours in the sound which are perceptually non-existent (phantom spikes). With psycho-acoustic experiments the number of spikes threshold from which a human cannot distinguish between a set of single phantom spikes and noise could be measured and used to avoid this effect.

Comparison to other Filterbanks We decompose sound as a linear combination of Gammatone functions as atoms. In contrast to the Short-Term Fourier Transform (STFT) with constant bin widths, we use filter bandwidths according to the percep-

tual ERB scale that increase exponentially with frequency. The Mel frequency scale used by MFCC's is spaced in a way similar to the ERB scale. Both the STFT and the MFCC's are calculated with a constant frame length in the temporal domain. For our approach, the length of the temporal filter decreases and the bandwidths of the filters increase with increasing center frequency and vice versa. Instead of plain sine functions the atoms in our decomposition are biologically plausible Gammatone functions, similar to Solbach et al.'s [66] wavelet filterbank built from Gammatone filters.

Relation to Other Matching Methods Let us discuss our approach in relation to three other matching methods, namely the Bag-of-Frames Distance, the Wasserstein-Mallows Earth Mover Distance, and Dynamic Time Warping.

Usually, the Bag-of-Frames Distance (BoF) [4] does not consider the time as a dimension explicitly encoded in the representation. Therefore, a sound is represented in the same way as the same sound played backwards, eventhough these are perceptually very different. They would yield a high dissimilarity in our measure. From our sparse representation of a sound it would be more difficult to determine the probability density than it would be using all full-dimensional feature vectors like it is done in the BoF Distance. The BoF Distance does not explicitly consider the difference between two features. It just gives a density estimate, thereby measuring if certain values happen equally often.

The method presented in this chapter can be considered as a approximation of the Wasserstein-Mallows Earth Mover Distance (EMD) [48]). The Hungarian algorithm we used here is a special case of EMD, where the optimal flow matrix is a 0-1 permutation matrix, i.e. the earth is only moved between two points. A binary value is assigned to the mass if a triple of frequency, time and amplitude is considered to be among the selected n most prominent spikes (mass 1) or not (mass 0). A variant would be to consider the amplitude as the mass in the EMD. Due to the sparsity of our representation of the sound and the 1-to-1 mapping between spikes (contrary to a soft-assignment proposed in the EMD) the computational expense of our method is significantly reduced in comparison to EMD.

Dynamic Time Warping (DTW) [63] is another matching method aligning two time series. In order to align two sounds, DTW maps the temporal evolution of one sound onto the time course of another sound. In contrast to our approach, which relies on a three-dimensional spike code, considered as a point graph, DTW is a non-linear but monotonous method that can match several time steps (samples) of one sound to one single time in the other sound. In order to perform this alignment, the spike code can be converted into a two-dimensional list and the spikes within this list can be sorted, according to some criteria. Hence, our method preserves structure that may be lost when applying DTW.

2.2.6 Conclusion

Following the everyday listening phenomenon, we have realized IRCAM's taxonomy of everyday sounds on two levels of description, the sound generating materials on the top level and the interactions between them on the bottom level. We exemplified these

sound categories with recordings of isolated sound events taken from the Sound Ideas database. Thereby we replicated a modification of IRCAM's multi-level taxonomy of everyday sounds based on the sounding physical interaction between materials.

We introduced a generic sparse sound coding scheme to be used in visualisation and classification of the everyday sounds. This representation, the so called *spike code*, not only considers spectral and energy properties, but also temporal characteristics. The use of the Gammatone filters links the representation to biological and psychoacoustic findings.

Plotting the spike code yields a visualization, a salient skeleton or a cartoon of the sound, that can be used, e.g. for automatic score transcription for *musique concrète*. Such a transcription provides valuable information about the sound visually indicating the audible similarities and differences between mostly and / or partly similar sounds, e.g. sounds stemming from the same sound class. Hence, a prototypical spike code can be generated for a sound class, or a set of prototypes for the sub-categories of that class. Sounds can be classified relative to these class prototypes by incorporating clustering methods. Sounds can also be resynthesized from a spike code representation, making prototypes audible.

The audible similarities and differences easily observable on spike codes enable us to define a structure preserving distance function via a graphical method, yielding a dissimilarity measure for sounds. We extend sound classification from Euclidean distances between feature vectors to the more general scenario of non-metrical dissimilarities. The potential of the method has been demonstrated by predicting attributes of everyday sounds. We have employed a distance substitution kernel to account for these pair-wise dissimilarities of spike codes retaining their structural features. With the P-SVM, we introduced an analysis method in sound and music computing that can deal with a more general class of problems. A metrical distance is not always available. For instance, in psychological experiments, we often deal with pairwise comparisons that do not qualify as a metric. The P-SVM is an appropriate classifier to be used. In binary discrimination and detection scenarios our method performed promisingly well, better than or in the same range as state-of-the-art methods.

The spike representation is a new approach to represent sounds, offering several new possibilities to pursue in classification and visualization frameworks. Although high classification accuracy is yielded in predicting classes of a perceptually relevant taxonomy, we plan to improve the psychological relevance of our predictor with data from further psychological experiments on everyday sound categorization.

2.3 Timbre Feature Reduction for the Classification of Perceptual Relevant Categories of Everyday Sounds

[2]

2.3.1 Introduction

We compare three radically different approaches for representation of everyday sounds: 1) the widely used Delta Mel Frequency Cepstrum Coefficients (DMFCC's), 2) Timbre Feature Reduction, consisting of a comprehensive set of timbre descriptors subject to Recursive Feature Elimination, and 3) a generic approach of spectro-temporal Spike Code Matching. The comparison of the methods is performed with recordings of everyday sounds that have been labeled during a forced-choice listening test with 36 subjects, given a perceptually relevant set of sounds. During this listening test, we asked listeners to categorize basic isolated sound events as *liquid, gas, solid, electronic*, and *machine*, representing important categories related to a natural focus on how the sound has been caused. In an evaluation with the Support Vector Machine, Timbre Feature Reduction yields an accuracy of 77.8 % compared to 66 % with DMFCC's, and 63.6 % with Spike Code Matching. Recursive Feature Elimination informs us which features are most important for the classification. Visualization of the reduced feature space by Fisher's LDA can be utilized as an intuitive visualization tool for sound design and control.

2.3.2 Representation and Classification of Sounds

Mel Frequency Cepstrum Coefficients Being a widely used method, we apply Mel Frequency Cepstrum Coefficients (MFCC's) [15, 49] as a reference. MFCC's 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. We use 13 MFCC coefficients. A feature vector is computed by taking the mean, variances, deltas between consecutive MFCC coefficients and the variances of these deltas over all frames, adding up to one 52-dimensional vector for a sound example (DMFCC's).

Timbre Descriptors As a comparison, we use the large set of timbre descriptors proposed by Peeters [55]. Including the DMFCC's, this set consists of several groups of features, temporal, energy, spectral, harmonic, perceptual, and other features. Whereas the harmonic features rely on frequency analysis and subsequent decomposition into sinusoidal partials, the perceptual features comprise descriptors related to perceptual dimensions of the timbre, like brightness. On the whole, we consider 166 features, several of them multi-dimensional (e.g. DMFCC's), resulting in 465 entries in the final feature vector. As an implementation we use the IRCAM descriptor [55] and the program pm2

[2] *from:* Robert Anniés, Olivier Houix, Hendrik Purwins,Nicolas Misdariis, Kamil Adiloğlu, Antoine Minard,Klaus Obermayer: Timbre Feature Reduction for the Classification of Perceptual Relevant Categories of Everyday Sounds, Manuscript in preparation for Speech Communication

to do spectral analysis. *Spectral* descriptors are calculated on the spectrum, *harmonic* descriptors only on the harmonics resulting from sinusoidal modeling, and *perceptual* descriptors are calculated on the 24 Bark bands. Specific loudness is calculated in 24 Bark bands. 12 MFCC's are calculated (in addition to the energy).

2.3.3 Perceptual Labeling of the Everyday-sound Database

Based on previous work on taxonomy of everyday sounds (see Section 2.3.1), sounds have been collected for a listening experiment to label sounds by classes of sound events. We focused on four big classes of sound events, based on the results of previous experiments [39] [47], i.e *liquid, solid, gas, machine*. A fifth class corresponding to *electronic* sounds has been added. The aim of this listening experiment is to build a collection of sounds for the classifier, that have been labeled with these five categories. The listening experiment was conducted in two phases: a first experiment has been conducted and toped up with a second experiment to provide enough sounds for the classifier.

Stimuli

All the sounds were recordings of activities occurring in various locations, chosen among different commercial sound libraries : Hollywood Edge Premiere Edition I, II and III, Sound Ideas General Series 6000, Blue Box Audio Wav, SoundScan and Extrem Fx. They were available with 16-bit resolution and a sampling rate of 44.1 kHz. The sounds correspond to sound events caused by interaction of different materials like: liquid, solid, gas, by machines as motor machines or electric machines or by electronic sounds without physical cause.

The sounds have been recorded with different techniques: near field and far field recordings. Experimenters have adjusted the loudness of the sounds to minimize variations of loudness between them.

Solid The class of *solid* sounds is related to sounds that can be produced by interaction of two solid materials caused by impact, friction and deformation due to mechanical contacts of one or more solid objects. *Impact* sounds are short sounds resulting from single instantaneous contacts between two objects or parts of one object. For example, we selected recordings of *switch, object dropping, coin falling, glasses shocking, dispenser coil*, and so forth. Concerning *friction* sounds, all such sounds involve an enforced contact of two objects moving against each other, for example, *rocks moving on a surface*; *bag zipper*, *knife scraping*, *fingernail scraping* ... These sounds correspond generally to continuous and regular acoustical pattern. The *deformation* sounds correspond to interaction between soft material like *crumpling a plastic bag*, or hard material like *glass breaking* or *cracking bone*. These sounds correspond to irregular acoustical pattern of micro events.

Liquid Physically, liquids do not emit much sound themselves. Still there are a lot of everyday sounds that are typical for liquids and caused by water indirectly.

The different selected *liquid* sounds correspond to four types of sounds: *drip*, *flow*, *pour* and *splash*. The *drip* sounds consist of recordings of dripping water. All examples have multiple countable drips (e.g. from a tap) or uncountable drips. The recordings also differ mainly in the reverberation of the room and the material type (liquid itself or solid), where the drips are reflected or absorbed, i.e. at the point of sound emission. Hence the distinguishing factor of the *drip*-sounds to the other *liquid* sounds according to acoustical description of sounds are small distinct acoustical events corresponding to drips of water falling vertically onto a surface. The *flow* sounds consist of examples of running water taps, waves on the shore or rivers and similar movements of water. These recordings encapsulate movements of portions of water that create swirls of air causing sound. The third type of *liquid* sounds is called *pour* sounds. Here we collected sound recordings of the interaction of transporting a portion of water from a vessel A to a vessel B through air, for instance filling a glass from a bottle. Sound is emitted form air bubbles that appear during the action. This definition seems close to that of the *flow*-sounds, but we have selected only mid-sized amounts of water/drinks (more than a drip, less than wave) and there must be a clear directed relocation of the water or drink from A to B. Because there are also sparkling drinks involved, the sounds are a mixture of the pouring action and the sounds that comes from the carbonated drink itself.

Gas Aerodynamic sounds sources are more direct. They create the sound by changing the atmospheric pressure. This can happen suddenly or as a steady process. The former are explosive sounds that populate the *explosion* category. *Gun shots, fireworks, cork*, and larger *detonations from TNT* were selected. Explosions result in a very energetic short bang, that is almost a Dirac impulse. Strong and long reverb from detonations is also a typical feature. The *crackling* sounds from fire recordings are another type of gas sounds that we selected. As *wind* we considered sounds resulting from a constant movement of air like recordings of *air pump*, *spray* .... The third type of sounds is labeled as *whoosh*. It differs from *wind* sounds in its transient characteristics. These sounds are caused by transient air blows (e.g. *flame thrower*) or objects moving fast through the air with relative velocity to the listener (microphone) with respect to the sound source.

Machine The machine sounds come from recordings of motor machines like *motorbike*, *train*, and *bulldozer* ..., and of electric machines, such as *electric saw*, *drill*, *industrial machine*, *food processor*, and *electric shaver* .... All these sounds are characterized acoustically by a relatively strong noise component, a continuous temporal envelope, regular or repetition patterns.

Electronic The *electronic* sounds we selected are produced by a device without a mechanical cause like *phone dial*, *cellular ring*, *car alarm*, or *digital timer*. Generally, these sounds are tonal and follow regular patterns.

Selection of Sounds For the first experiment, 213 sounds have been selected, corresponding to six expected classes: 10 *electric* sounds, 49 *gas* sounds, 33 *liquid* sounds, 17 *motor* sounds, 92 *solid* sounds, 12 *electronic* sounds. The *electric* class was not retained for the second experiment due to confusion with the *electronic* class. Concerning the second experiment, we chose 54 *machine* sounds and 45 *electronic* sounds and also sounds from the fist experiment (11 *solid*, 9 *liquid*, 10 *gas*) to be in a similar experimental setup.

This repartition of sounds through different classes is not the result of the experiments, these are expected classes, e.g. derived from the labeling in commercial data bases.

Participants

20 participants (11 women and 9 men) volunteered as listeners and were paid for their participation for the first phase. For the second phase, 16 participants (8 women and 8 men) from IRCAM have participated. Participants were not necessarily musicians.

Apparatus

The sounds were played by a Macintosh Mac Pro (Mac OS X v10.4 Tiger) workstation with a MOTU firewire 828 sound card for the first phase and a Fireface 800 for the second phase. The stimuli were amplified diotically over a pair of YAMAHA MSP5 loudspeakers. Participants were seated in a double-walled IAC sound-isolation booth. The experiment was run using the PsiExp v3.4 experimentation environment including stimulus control, data recording, and graphical user interface [64]. The sounds were played with Cycling'74's Max/MSP version 4.6.

Procedure

The experiments were performed by the IRCAM group. Participants had to read the instructions first. They were explained that they would have to listen to sounds recorded in different situations. They were asked to identify the physical action causing the sound. Before starting the experiment, participants listen to a representative sound sequence composed by 25 sounds from different classes of sound events. They have to choose between five classes corresponding to:

- *gas* like *explosion, blowing*, ...,
- *liquid* like *bubble, flowing*, ...,
- *solid* like *shock, crushing, crumpling*, ...,
- *electronic*,
- *machine* like *motor, electric*.

They were allowed to listen to each sound many times and were asked not to counterbalance their answers between classes. In the first experiment, a sixth class was introduced, *electric*, but removed for the second experiment because of the few number of sounds associated to it and a possibly confusing definition with respect to the *electronic* class.

classified as	g.	l.	s.	e.	m.
31 gas sounds	28	0	3	0	0
31 liquid sounds	0	31	0	0	0
81 solid sounds	0	0	81	0	0
51 electronic sounds	0	0	0	51	0
59 machine sounds	3	0	0	1	55

Table 2.3: Number of sounds classified as *gas, liquid, solid, electronic, machine* during the listening experiments. Here, only sounds are shown that have been assigned with $>$ 60 % to one particular class. It can be observed that sounds that were expected to be labeled as gas sounds were perceived as solid sounds, and some sounds expected to be perceived as machine sounds were labeled as gas or electric sounds.

Analysis

The collection of labeled sounds used by the classifier is based on the results of two complementary experiments of a 5 alternative forced choice test. From the first experiment, we removed the sounds that have been classified as *electric*, even if only by one subject. We kept the data of each of the 167 remaining sounds classified as *electric, gas, liquid, machine, solid*. From the second experiment, we yielded 99 additional sounds. For each of the two data sets, the number of responses for each sound within the five classes has been transformed to frequencies. After that, the results of the two experiments have been merged into a collection of 266 sounds classified according to frequency as *electric, gas, liquid, machine*, and *solid*. 253 sounds have been classified with a minimum of 60% within a class (see Table 2.3).

2.3.4 Feature Selection, Support Vector Machine Classification, and Dimension Reduction

First we extract a large set of timbre features ([55], Section 2.3.2) yielding overall 465 features. We normalize the data mapping them linearly to the interval $[0, 1]$ and setting the missing values (for some harmonic features) to the mean. We use a support vector machine with radial basis functions for classification. Through 10-fold cross validation, we optimize the parameters γ (width of basic functions) and C (penalty for error term). For each training partition, we select the 20 best features with Recursive Feature Elimination [33]. These features are then used as an input by the support vector machine for testing.

For feature extraction we use IRCAM's pm2 and the IRCAM descriptor [55]. For grid search and support vector machine, we apply Weka [78] in combination with the LibSVM [8].

In order to understand which features are relevant for the discrimination between sound classes, we use Recursive Features Elimination on the entire data set and discuss the results. In addition, we use Fisher's LDA for visualization, in the implementation of the Matlab toolbox for dimensionality reduction [74].

Representation	Accuracy (%)
DMFCC's	66.0
TF	71.8
TFR	**77.8**
Spike192	63.6
Spike160	62.0
Spike128	60.0
RG	20.0

Table 2.4: We compare classification results for various representations. The accuracy for the 5-class numerical experiment is shown. The following methods are employed: MFCC's including their deltas (DMFCCs), the full timbre feature set including DMFCCs (TF), 20 features resulting from automatic Timbre Feature Reduction (TFR), Spike Code Matching with different numbers of spikes (Spike192, 160, 128), and random guess (RG). A support vector machine with 10-fold cross validation has been used. Timbre Feature Reduction performs best.

2.3.5 Results

Classification Comparison

The unnormalized accuracy in the multi-class setting serves us to evaluate and compare Timbre Feature Reduction with the widely used DMFCC's, the full set of timbre features and the Spike Code Matching with various numbers of spikes. For all methods except for Timbre Feature Reduction we used grid search to determine the best parameters for the support vector machine with radial basis functions. For Timbre Feature Reduction, we took the parameters of the support vector machine that yielded best results on the timbre feature subset extracted from the entire data set ($\gamma = 0.0078125, C = 256$, see Section 2.3.5). The timbre descriptor results are based on 248 out of 253 sounds, since 5 sounds could not be processed by the timbre descriptors, since they were too short or for other reasons the descriptors could not be applied in a meaningful way. Table 2.4 reveals that Timbre Feature Reduction outperforms Spike Code Matching with 192 spikes by 15.8 % and DMFCC's by 11.8 %. As a comparison with a base line, random classification yields 20 %.

Delta MFCC's generally perform well. They are able to capture characteristics of short sound bites by taking into account the mean and variance of the original signal as well as its delta, therefore characterizing a sound by its lower statistical moments in phase space. Spike Code Matching is biologically plausible in two ways: it decomposes the sound into biologically plausible atoms, the Gammatone functions, and it produces a sparse representation of a sound. In addition, it establishes a one-to-one correspondence of spike representations between two sounds, yielding pairwise sound dissimilarities, thereby explicitly encoding the temporal structure of the sound. For our data set, the results indicate that psychoacoustic representation schemes, such as the timbre descriptors are able to capture the complexity in the sounds better than the generic methods

(DMFCC's and Spike Code Matching).

Selected Features From the 465 timbre features, we remove features by Recursive Feature Elimination on the entire data set until only 20 features remain, in order to extract features relevant for our classification problem. We yield the following subset: (For descriptors extracted from a filter bank the filter bank index is indicated.)

Noise energy	*Percept. spectral variation*	*DDMFCC var 6*
DMFCC var 3	*MFCC var 1*	*Rel. specific loudn. 24*
Harm. spectral variation var	*Rel. specific loudn. var 15*	*Noisiness*
Total energy	*MFCC var 3*	*MFCC var 4*
Harm. tristimulus 1. band	*Fluctuation strength*	*Spectral spread*
Spectral variation	*Length*	*DDMFCC var 4*
DMFCC variance 11	*Percept. spectral skewness*	

In this list, we consider the means of the descriptor calculated in frames over the length of the sound if no other specification is given.

Interestingly, noise related features are prominent in this selected feature list (*noise energy, noisiness*). This may explain the superior classification by Timbre Feature Reduction because noise may not be represented adequately in Spike Code Matching nor in DMFCC's. Although the stimuli have been calibrated beforehand, the *total energy* plays a role. *Fluctuation strength* and sound *length* are also important. It is difficult to interpret the meaning of the selected various components of multidimensional spectral representations (*DDMFCC variance 6, DMFCC variance 3, MFCC variance 1, relative specific loudness 24, relative specific loudness variance 15, MFCC variance 3, MFCC variance 4, DDMFCC variance 4, DMFCC variance 11*). The specific loudness analysis emphasizes particular bands, in our case bands 15 (middle range) and 24 (highest frequency band). The MFCC variances emphasize the variability in particular bands, in our case bands 1 and 4, the variation of the deltas of bands 3 and 11, the second deltas of band 4 and 6.

The frequent appearance of the variance indicate the importance of statistical moments as a global sound characteristics. Several occurrences of the deltas and double deltas underline the significance of the derivatives to describe the sound, especially since no MFCC mean values are selected. *Harmonic spectral variation, perceptual spectral skewness variance, spectral spread variance, spectral variation* all characterize the variability of the spectral envelope.

Visualization In Figure 2.4, we apply Fisher's LDA on the 20 features that remained after Recursive Feature Elimination. For a detailed analysis, we examine how the original features contribute to the four main components extracted by Fisher's LDA. This indicates that the data distribution is characterized by the interrelation between *harmonic tristimulus (1. band), total energy, noise energy*, and *fluctuation strength*. The first component of the LDA is dominated by the 1.band of the *harmonic tristimulus* (-61.4), to a lesser extend by *total energy* (-32.7), *fluctuation strength* (-20.6) versus *noise*

energy (20.8). The second component is characterized by the contrast between *noise energy* (-84.4) versus *total energy* (24.7). The third component is defined by *harmonic tristimulus (1. band)* (42.6) and *noise energy* (38.2) versus the *total energy* (-26.0). *Harmonic tristimulus (1.band)* (60.6), *total energy* (33.1), *perceptual spectral variance* (26.3) versus *DDMFCC variance* 6 (-18.3) and *fluctuation strength* (-18.0) constitute the fourth component. In Figure 2.4, we see how the classes *electronic* and *machine* separate well in this projection, whereas the classes *solid, liquid, gas* overlap.

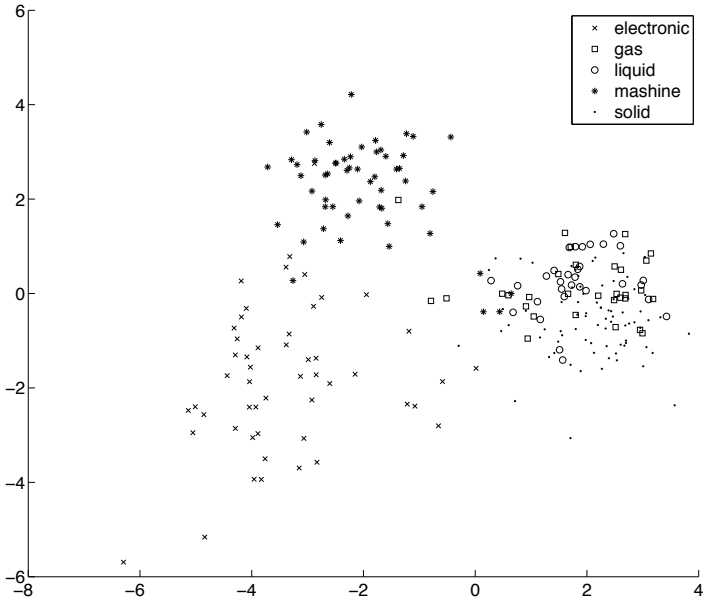


Figure 2.4: Fisher LDA of the 5 sound classes. We observe that the classes *electronic* and *machine* are clearly separated from the other classes. Certain parts of the *solid* class are separated, whereas others mix especially with *gas* and to a lesser extent with *liquid*.

2.3.6 Discussion and Conclusion

The timbre descriptors mimic psychoacoustic evidence, thereby constituting a mid-level description level that is intelligible. Feature extraction on the timbre descriptors can be interpreted as a way to approach the saliency problem in sound and music research. The effectiveness of Timbre Feature Reduction is supported by the comparison with the widely used DMFCC method. In comparison to DMFCC, Timbre Feature Reduction improves the accuracy by 7.3%.

Timbre Feature Reduction prepares the ground for interactive sound search and visualization. Practically, it can aid sound design [68], sound track composition for movies as well as control of sound synthesis when combined with a regression method to map this intuitive sound space to the space of sound synthesis parameters.

It is difficult to compare classifier performance for everyday sounds due to the lack of an established and widely accepted sound taxonomy and corresponding benchmark data sets. In our case, data has been selected in a way to capture a large range of sound phenomena. Therefore, our data base contains a great variety of sounds and a relatively small number of instances per class, which makes automatic learning of the classes especially challenging.

In Table 2.3, the difference between the labels in commercial sound databases and the perceived causes of the sounds indicate a discrepancy. There is a perceptually crucial but subtle difference between how a sound is produced and how a person perceives how the sound may have been produced. We validated our model by a perceptual experiment. Often, in research in sound and music retrieval, the classifier performance is evaluated by a label set that is either derived from the way the sound (or the music) has been produced (the true sound source) or by a manual labeling procedure. Such a labeling procedure is not only biased by the subjective listening of one person and prone to errors if not cross-checked by several people. But also, the labeling is biased if it allows e.g. the use of graphical sound analysis tools. In this case, the labels do not adequately present the perception of sound in a real listening situation.

The scope of Timbre Feature Reduction is not limited to the domain of everyday sounds. It can also be applied to other sound and music analysis tasks [58, 35]. The method flexibly adopts the individual user's viewpoint expressed by their particular discrimination or preference.

2.3.7 Acknowledgment

We would like to thank Patrick Susini and Guillaume Lemaitre for inspiring discussions, Juan Jose Burred and Geoffroy Peeters for help with the programs IRCAMdescriptors and mp2.

3 Predictors and Visualization for Emotional Attributes

[1]

3.1 Introduction

Altering sounds to evoke a specific emotion in the listener would be an outstanding tool for sound design in, for instance, consumer behavior, movie sounds and product design. A first step towards this ambitious goal is to be able to predict emotional attributes related to sounds, a procedure that will be explored in this chapter. Previous research has investigated emotional prediction in music, however this has not yet been done for sounds. By means of descriptors originally designed for music and speech, features are extracted from sound data and are used to teach a support vector machine to recognize emotions in sound. In the first part of the numerical experiment, features are extracted from short selections from emotionally labeled music. The second part will use the same procedure on everyday sounds.

With sounds affecting the emotional state of human beings significantly, one can imagine how knowledge on the relation between acoustical and emotional features of sounds contribute to fields such as consumer behavior and moviemaking, especially, if these emotional reactions to sounds can be predicted by a machine. For a detailed account on emotions in everyday sounds see [38]. Based on machine-learning approaches used for music genre classification [71] [32] and mood detection in music [23], [45], we will attempt to train an emotional classifier for sounds. Features that have an effect on the emotional labeling are extracted and a SVM is subsequently taught to automatically classify sounds into an emotional class. Since an emotional reaction can be elicited in a segment of music as short as one second [5], the feature extraction and SVM training procedure will first be performed on short music selections (1 to 4 seconds in length). As well as being a first approach towards training a SVM for non-musical sounds, this approach would give information in the field of sound design. The second part of the numerical experiment will perform the feature extraction and SVM training on non-musical everyday sounds. Compared to emotions in music, emotions in everyday sounds are predominantly related to their context [26]. However, also when considered out of context sounds can evoke emotions [76]. The psychological mechanisms underlying emotions to sounds in comparison to music should be taken into account [42]. Emotions based on arousal and valence as a reaction to acoustic features of sounds would

[1] *from* Sara de Bruijn, Hendrik Purwins, Robert Annies, Cyril Laurier, Kamil Adiloglu, Klaus Obermayer: Comparative Prediction of Emotional Attributes of Sound and Music Bites, Technical Report

be possible at the most basic level, a mechanism Juslin & Västfäll label as the brain stem reflex. This mechanism can explain the arousal and pleasantness based solely on the most basic acoustic features of sound. In Russell's circumplex model[62] the valence and the arousal dimension span the emotional space. As an example, high arousal is linked with aggressive (low valence) and happy emotions (high valence). In the case of low arousal, we can consider sad (low valence) and relaxed (high valence). This is a very simplified way of categorization, where emotions within one of the four categories may be quite different. However, this model is considered adequate [45], keeping in mind its limitations.

3.2 Numerical Experiments 1

This part of the numerical experiment explores the emotional content of short music fragments. Features that have an effect on the emotional labeling are extracted using timbre descriptors and a SVM is subsequently taught to automatically classify short musical sounds into an emotional class.

3.2.1 Data Labeling

Short fragments of music (2 to 4 seconds in duration) are manually extracted from songs that have been manually annotated in categories happy, sad, aggressive and relaxed. Half of the data used is from a database which consists of songs pre-labeled by a large online community (Last.fm) according to the listeners choices. Subsequently the relevantly labeled songs were again validated by experts according to the relevant labels [46]. The other half of the data is extracted from movie tracks annotated by music students [20]. Significant features are extracted and a classification of features is made for each emotion.

3.2.2 Timbre Descriptors

As a comparison, we use the large set of timbre descriptors proposed by [34]. Including the MFCC's, this set consists of several groups of features, temporal, energy, spectral, harmonic, perceptual, and other features. Whereas the harmonic features rely on frequency analysis and subsequent decomposition into sinusoidal partials, the perceptual features comprise descriptors related to perceptual dimensions of the timbre, like brightness. On the whole, we consider 166 features, several of them multi-dimensional (e.g. MFCC's), resulting in 465 entries in the final feature vector. As an implementation we use the IRCAM descriptor [34] and the program pm2 to do spectral analysis.

3.2.3 Observations

The emotional content of the 2 to 4 second selections were similar to the content of the song from which it had been extracted. This is especially the case for the aggressive and happy music. These categories were distinguishable from the low-arousal categories (sad and relaxed). The distinction between sad and relaxed selections was observable

for the movie soundtracks, but is less evident in the selections taken from the Last.fm data-base. This could be the result of not including vocals in the analysis, since vocal acoustic information and lyrics [55] contribute to the valence of the excerpt [20]. The selections corresponding to the aggressive cluster are overall high in tempo, loud and sound chaotic. Overall, the happy selections are high in tempo and consonant. All selections are in major mode. The overall sound of the relaxed selections varies, but overall the tempo is low and in some selections there is no rhythm. The sad selections are similar to the relaxed, but are loaded with a negative valence due to the minor mode. However, in the case of high energy, the sound is a dramatic kind of sad, which might be closer to angry than sad music. The movie tracks seem to convey more sadness than the selections extracted from popular music. The instrumentals of the movie tracks could be giving information that would have been given by the (omitted) vocals of the popular tracks. Intuitively, the most evident features to make the distinction based on the axes valence and arousal would be tempo and mode. The mode could be extracted and presumably will be able to make the distinction between positive and negative valence. Tempo can make the distinction between high and low arousal. If it is possible to get the information from these short music fragments, articulation could be a predictor of the valence of the selection.

3.2.4 Results

The following selection of features [55] is based on the 20 most significant features according to each the SVM Attribute Evaluation, the Info Gain Attribute Evaluation and the Cfs Subset Evaluation [34], computed in WEKA [77].

Features Related to Noisiness:

- Spectral Flatness mean (mostly 3, also 1, 2, 4)
- Spectral crest mean

Perceptual Features:

- Relative specific loudness mean (Most importantly bands 14, 15, 16, 17, 18 and 24, but also bands 7, 10, 19, 20, 21, 22 and 23).
- Relative Specific Loudness variance (7)
- Perceptual Spectral Variation variance
- Perceptual Spectral Kurtosis variance
- Loudness mean
- Perceptual Spectral Skewness mean
- Perceptual Spectral Kurtosis mean - Spread mean

Spectral Features:

- Spectral Variation mean (1, 2)
- MFCC mean (1)
- MFCC variance (4, 12)
- DDMFCC mean (4)
- DMFCC variance (mostly 2, then 1,8,10)
- Spectral Spread mean
- Spectral Centroid mean
- Spectral Kurtosis mean
- Spectral Crest mean

Harmonic Features:

- Odd To Even Harmonic Ratio mean (3)
- Harmonic Spectral Spread variance
- Harmonic Spectral Spread mean
- Fundamental Frequency Modulation
- Amplitude value
- Harmonic Spectral Skewness mean

Energy Features

- Energy Modulation Amplitude value (1)

Instantaneous Temporal Features

- Signal Auto Correlation variance (3)

3.2.5 Discussion

Noise features: The selected features related to noise are Spectral Flatness mean 1, 2, 3, 4 and the Spectral crest mean 3, 4. Spectral flatness is a measure of the tonality of the selection. An SFM value closer to 0 represents tonality closer to 1 (both measures on a scale from 0 to 1). Whereas for dissonant sounds the SFM value would be closer to 1 than to 0. SFM descriptors of the frequency range 250-5000 Hz were all selected as relevant attributes.

Classified as	A	H	R	S	Sum
Aggressive	42	7	0	2	51
Happy	2	39	4	3	48
Relaxed	0	4	29	17	50
Sad	1	6	9	34	50

Table 3.1: Confusion Matrix for Music.

Perceptual Features: The perceived loudness is described with the Relative specific loudness means. These features will probably distinguish between high and low arousal music selections. These means were measured on the Bark band scale, which gives the best approximation of the human auditory system. For the relative specific loudness mean, the Bark bands 14, 15, 16, 17, 18 (2000-3700 Hz) speech-like, harder sounds, clarity in guitar and piano) and 24 (12000-15500 Hz) cymbals, brilliance) are considered most important. The bands 14-18 could represent harder sounds, speech-like or for instance as a clear guitar or piano sound. Band 24 is more related to brilliant sound and cymbals for example. Bands 7 (630-770 Hz), 10 (1080-1270 Hz), 19, 20, 21, 22 and 23 (frequency range 4400-9500 Hz) are also considered descriptors. The descriptors based on the variance of the Relative Specific Loudness were based on Bark band 7 (630-770 Hz). Thus within the sound segment, variations on this frequency band are most descriptive of distinguishing emotionally labeled sounds.

Spectral features: The spectral shape features are used to describe the sound. From the sound descriptions it seems that the happy sounds contain the highest tones in comparison to relaxed, sad and aggressive. The spectral features possible also may describe typical sounds from the aggressive selections, such as the distortion and also often high pitched sounds.

Harmonic features: Odd To Even Harmonic Ratio mean (3) would be a way to describe the sound of the selection. In the case of aggressive selections this feature could be quite different in describing the prominent guitar sounds than the piano sounds in the sad and relaxed sound selections. The Energy Modulation Amplitude value in combination with Fundamental Frequency Modulation Amplitude value could provide a description of the sustained notes typical for the sad and relaxed music selections. As in the mentioned examples, the instruments used in these selections are mainly natural (in the examples guitar, piano and violin) and seem to contain much of the expressive energy, as these features potentially could describe.

3.3 Numerical Experiments 2

This part of the numerical experiment explores the emotional content of everyday sounds, such as door sqeaking, explosions, sea waves, water bubbling, bell, high speed car, gun shots. Sounds are subjectively labeled as being aggressive, happy, relaxed or sad. The

sounds are chosen so that the characteristics of the aucoustic signal should imply emotional content that is not only due to the semantics of the sound. Features that have an effect on the emotional labeling are extracted and a SVM is subsequently taught to automatically categorize short musical sounds into an emotional class.

3.3.1 Data Labeling

Sound excerpts of 1 to 3 seconds were extracted from commercial sound databases. The excerpts were classified as either aggressive, happy, relaxed or sad. These were again blindly rated by a second subject to correct for ambiguity. Features were then extracted from the sounds that were rated similarly by both subjects.

3.3.2 Results

The following selection of features [34] is based on the 20 most significant features according to each the SVM Attribute Evaluation, the Info Gain Attribute Evaluation and the Cfs Subset Evaluation, computed in WEKA [77].

Features Related to Noisiness:

- Spectral Flatness mean (2,3,4)
- Spectral Flatness variance (1)
- Spectral Crest variance (4)
- Noisiness mean

Perceptual Features:

- Relative Specific Loudness mean (2, 3, 8, 13, 16, 18, 19, 20, 21, 23, 24)
- Relative Specific Loudness variance (21,23,24)
- Perceptual Spectral Variation variance
- Perceptual Spectral Spread mean
- Perceptual Spectral Variation mean
- Loudness mean
- Band Spectral Deviation mean

Spectral Features:

- Spectral Variation variance
- MFCC mean (1)
- DMFCC mean (1)
- DMFCC variance (1)
- DDMFCC mean (3)
- Spectral Centroid mean
- Spectral Variation mean

Harmonic Features:

- Odd To Even Band Ratio mean
- Harmonic Energy mean
- Harmonic Spectral Variation mean
- Harmonic Spectral Slope variance
- Roughness

Energy Features

- Total Energy mean
- Noise Energy mean

Global Temporal Features

- Temporal Increase

3.4 Classification

The LibSVM is used to predict the label the given sound selections. The following algorithms were performed after normalizing the data. A 10-fold cross validation via grid-search was used to select the parameters. Various simulations have been performed, e.g. with radial basis functions as kernels or SVM with previous feature selection. We employed grid search to find the best parameters for the SVM. The best accuracy was obtained at a cost term of 1.1. The linear kernel type gave the most accurate results. The gamma parameter is not of use with this kernel type. For the complexity term C=1.1, the LibSVM gave a classification accuracy of 72.4%. The confusion matrix for the music selections (table 3.1) and sounds (table 3.2) give a clear view of the classification.

For the same parameters we yield a classification accuracy of 74.3

Classified as	A	H	R	S	Sum
Aggressive	31	0	1	3	35
Happy	0	18	2	1	21
Relaxed	3	4	16	1	24
Sad	5	3	3	10	21

Table 3.2: Confusion Matrix for Sounds, For sad the accuracy is lowest. For aggressive best.

3.5 Visualization

The Matlab visualization toolbox [73] was used to facilitate the representation of the data by means of dimensionality reduction. The data was represented in a 2 dimensional space by means of different dimensionality reduction techniques. The data seemed well reduced by means of the Isomap [70] 3.1.The number of data points represented in the lower dimensional representation was reduced from 199 to 178. This is because Isomap only embeds the largest connected component in the neighborhood graph [73]. A cluster distinction could be made between the low arousal (relaxed and sad) and high arousal (aggressive and happy) data. For the low arousal data, the valence seems to be indistinguishable. The aggressive data seems furthest separated from the rest of the data. The happy data seems to overlap mostly with the aggressive data, however is less distinguishable.

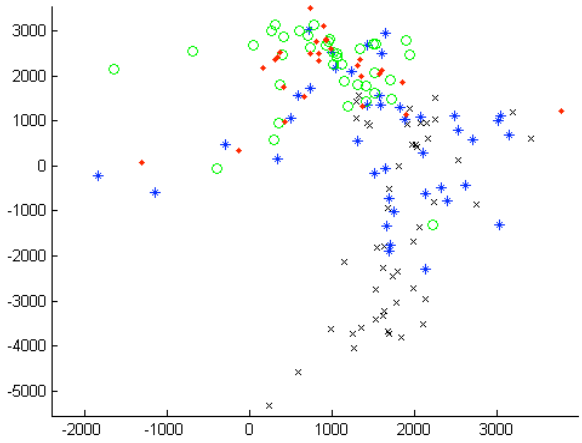


Figure 3.1: Visualization of everyday sounds by means of Isomap. Emotional labels are indicated by color (green ○ = relaxed, red += sad, black x = aggressive, blue * = happy). This is a zoomed view, 21 outliers are not in view. A segregation can be made between low (relaxed and sad) and high (happy and aggressive) arousal and to a certain extent between aggressive and happy.

3.6 Discussion

A difficulty concerning classification of emotions to sounds is the ambiguity of defining emotion. While for music selections and everyday sounds the same categories were used, it is not clear whether these categories really represent the same emotion. In order to describe everyday sound with an emotion based on the acoustic features, the context had to be omitted from the selection. However, from the double classification of the sounds it seemed that in some cases this is not completely realized.

4 Visualization Techniques

4.1 Generating Audio Prototypes

[1] Prototypical sounds of categories are artificial sounds that have the smallest average distance to a all sounds in a category. They can play the role of defining a category and show the typical features. The notion of a distance between sounds is essential to find prototypes. Machine learning theory knows different forms of calculating or estimating prototypes using a set of training examples. In the following study the technique of pairwise clustering is used using the spike code representation [2], [65] of everyday sounds. The categories are the same 10 which were presented in section 2.2.1.

4.1.1 Study Goals

One of the objectives of this part of the CLOSED project is to eventually develop models for sound synthesis. Once such models have been established they can be optimized by using prototypical spike coded sounds from each category. However, the computation of such prototypical sounds is not trivial. For instance, the average of two spike coded sounds is the average of spike pairs between both sounds. Similar to when determining the distance between two sounds, a mapping of spikes from one sound to the other is needed. Therefore we will present a method that is able to generate sound prototypes by performing a spike mapping.

However, when looking at some characteristics of the sounds it seems that all 10 sound classes consist each of a certain number of sub-classes being characterized by consisting of sounds with high self-similarity. One evidence for this assumption arises when listening to the sounds. Subjectively one perceives sounds which are very similar among each other and dissimilar to others, for instance in terms of quality, duration or content. Another observation that corroborates this assumption is when surveying the visualized distance matrix that contains the distances between each sound within a sub-category.

Figure 4.1 shows such a distance matrix, in this case of the sounds in sub-class 'Wind'. The sounds were extracted from the sound database in alphabetically order and thus in this particular distance matrix they form groups of sounds that are very similar to each other. This is depicted by the blue squares along the diagonal, which at the same time show a high dissimilarity to sounds outside that group, depicted by the warmer color code of distances outside the squares. Both this observation and the subjectively perceived clusteredness within sub-category sounds suggest that it would be adequate to refine the hierarchical categorization by defining sub-categories within categories. We will present a pairwise clustering algorithm that uses distances between sounds to find an optimal

[1] from Matthias Schulze Kraft (MA student supervised at NI-BIT).

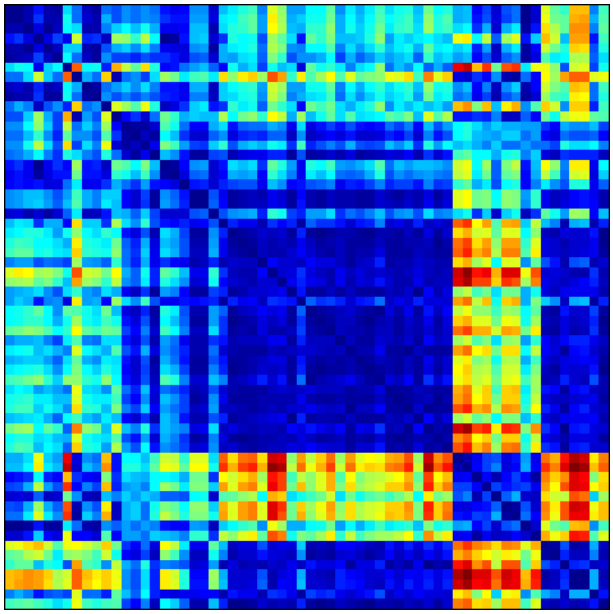


Figure 4.1: Visualization of the distance matrix of the sounds in sub-class 'Wind'. The color code goes from blue to red colors, dark blue means zero distance (e.g. the self-distances in the diagonal), red means big distance.

partitioning of sound classes into sub-classes. Furthermore, in order to determine the optimal number of sub-classes of sounds per class, we will formulate a criterion that allows to compare the results of clustering with different number of clusters.

4.1.2 Methods

Pairwise Clustering

The problem of optimally partitioning a data set depends on the representation of the data. When the data is given in vectorial form, for instance as feature space coordinates, an approach known as central clustering is applied which quantizes the data minimizing the quantization error. One well-known example for a central clustering algorithm is k-means. However, in some cases the feature space of the data set is either not known or of no interest and the only information at hand are pairwise comparisons of data objects. In such cases an approach known as *pairwise clustering* is applied. The pairwise comparisons consist of determining a distance measure that depicts the similarity of two objects. Formally, a distance matrix D_{ij}, with $i, j = 1, ..., N$ is constructed by applying the distance measure to a data set containing N objects. The idea now is to group the data to clusters such that the sum of distances between objects of the same cluster is minimized. Herefore an assignment matrix $M \in \{0, 1\}$ is introduced:

$$M_{i\nu} = \begin{cases} 1 & \text{if object } i \text{ is assigned to cluster } \nu \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The cost function for pairwise clustering with K clusters is then

$$\mathcal{H}^{pc}(M) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{D_{ij}}{N}\left(\sum_{\nu=1}^{K}\frac{M_{i\nu}M_{j\nu}}{p_\nu} - 1\right) \tag{4.2}$$

where $p_\nu = \sum_{i=1}^{N} M_{i\nu}/N$ normalizes the cost of a particular cluster ν.
Here, an algorithm for pairwise clustering was implemented that makes use of an optimization strategy know as *deterministic annealing* [36]. The key idea of deterministic annealing is to introduce a temperature parameter T that is gradually reduced during the optimization process. High temperature smoothes the cost function and low temperature reveals the full complexity (recovering the original cost function for $T \to 0$). Thus the system is forced into solutions with low costs and the probability to end in a local optimum of search space is lower compared to other optimization strategies. Applying deterministic annealing to the optimization problem of pairwise clustering implies relaxing the constraint on the assignment matrix M such that now $M \in [0, 1]$ and $\sum_{\nu=1}^{K} M_{i\nu} = 1, \forall i$.
One issue that arises when doing pairwise clustering is that the object assignments M are statistically dependent since each assignment variable $M_{i\nu}$ interacts with all other assignment variables. However, since these cost contributions converge to averages in the limit of large data, these averages can be approximated using the *mean-fields* $\mathcal{E}_{i\nu}$. Applying a variational approach and assuming zero self-distance ($D_{ii} = 0, \forall i$) the optimal mean-fields for assigning object i to cluster ν are given by

$$\mathcal{E}_{i\nu} = \frac{1}{\sum_{j\neq i}^{N} M_{j\nu} + 1}\sum_{k=1}^{N} M_{k\nu}\left(D_{ik} - \frac{1}{2\sum_{j\neq i}^{N} M_{j\nu}}\sum_{j=1}^{N} M_{j\nu}D_{jk}\right) \tag{4.3}$$

and result in the optimal assignment variables

$$M_{i\alpha} = \frac{exp(-\mathcal{E}_{i\alpha}/T)}{\sum_{\nu=1}^{K} exp(-\mathcal{E}_{i\nu}/T)}. \tag{4.4}$$

An algorithm that estimates $M_{i\nu}$ works as follows: After choosing a starting temperature T the algorithm proceeds with an inner and an outer loop. In the inner loop $\mathcal{E}_{i\nu}$ and $M_{i\nu}$ are estimated alternately according to (4.3) and (4.4) until the mean-fields converge. The outer loop iterates the inner loop, reducing the temperature gradually in each iteration, until some termination criterion is reached. A reasonable termination criterion is to let T become so small that all assignment variables in M are either 1 or 0, i.e. that the original constraint (4.1) is fulfilled.

Learning Object Prototypes

Gold et al. [29] presented a method that can solve the assignment problem. They used the presented technique to estimate coordinate transformations between 2D- or 3D-point sets and eventually calculate prototypes of those point sets. However, the measure they

use to calculate distances between points and hereby to estimate the transformations is exchangeable and thus a more general approach can be stated. Here we derive a generalization of that technique, which we call *element matching*, such that using the distance measure for spike coded sounds it can be used to compute sound prototypes. In a further work Gold et al. extended this technique such that it can also do a clustering on the point sets, i.e. the point sets are partitioned into clusters according to the distance between the point sets and their prototypes [30]. We also included this extension, which we call *object clustering*, into the generalized assignment method. We will now present the objective functions and the algorithms in a rather simplified way, for details see references.

Objective Function for Element Matching

Given two sets of elements $\{x_j\}$ and $\{y_k\}$ and a distance measure $d(x_j, y_k)$, element matching finds the best mapping of the elements in x to the elements in y such that the total distance between both sets is minimized, which is achieved by minimizing

$$\mathcal{H}^{em}(m) = \sum_{j=1}^{J}\sum_{k=1}^{K} m_{jk} d(x_j, y_k), \tag{4.5}$$

with the constraints $\forall j \ \sum_{k=1}^{K} m_{jk} \leq 1$, $\forall k \ \sum_{j=1}^{J} m_{jk} \leq 1$ and $\forall jk \ m_{jk} \geq 0$. The correspondence matrix m matches the elements of one set to the elements in the other set. The objective function $\mathcal{H}^{em}$ defines at the same time the distance between the two sets of elements $\{x_j\}$ and $\{y_k\}$.

Objective Function for Object Clustering

Given a set of I objects $\{X_i\}$ find a set of B cluster prototypes $\{Y_b\}$ and assignment variables M_{ib} defined as

$$M_{ib} = \begin{cases} 1 & \text{if object } X_i \text{ is assigned to } Y_b\text{'s cluster} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

such that the distance of all objects from the respective cluster prototypes is minimized. If the distance between objects X_i and Y_b is given by the distance measure $D(X_i, Y_b)$, $\{Y_b\}$ and M_{ib} are found by minimizing

$$\mathcal{H}^{oc}(Y, M) = \sum_{i=1}^{I}\sum_{b=1}^{B} M_{ib} D(X_i, Y_b) \tag{4.7}$$

The Algorithm

The following algorithm, analogously to the pairwise clustering algorithm, makes use of deterministic annealing as an optimization strategy. One annealing process, controlled

by the parameter β_m, is used to estimate the correspondence variables m_{ik} and the object distances and runs in an inner loop. The second annealing process, controlled by the parameter β_M, runs in the outer loop and estimates the assignment variables M_{ib} and the object prototypes Y_b.

Inner Loop The inequality constraints on m presented in 4.1.2 ensure that one element in x_j corresponds to at most one element in y_k. In order to allow null matches, for instance when $J \neq K$, slack variables are introduced into the corresponding matrix by adding an extra row and column and the constraints on m are transformed into the equality constraints $\forall j \sum_{k=1}^{K+1} m_{jk} = 1$ and likewise for the column constraints. Now in order to find the optimal element mapping a procedure is applied, which consists in the iterative process of alternating row and column normalizations. Initially a matrix Q_{jk} is constructed which contains the distances between every x_j and y_k, and a matrix m^0_{jk} is initialized with $exp(-\beta_m Q_{jk})$ and adding the slack variables. Now the inner loop runs the alternating row and column normalizations as follows:

$$m^1_{jk} = \frac{m^0_{jk}}{\sum_{k=1}^{K+1} m^0_{jk}} \tag{4.8}$$

for the normalization of all rows and

$$m^0_{jk} = \frac{m^1_{jk}}{\sum_{j=1}^{J+1} m^1_{jk}} \tag{4.9}$$

for the normalization of all columns; these alternating normalizations are performed iteratively until m^0_{jk} converges.

Outer Loop The outer loop proceeds in three phases: (i) the correspondence variables m and object distances are computed in the inner loop, (ii) the assignment variables M are estimated:

$$M_{ib} = \frac{exp(-\beta_M D(X_i, Y_b))}{\sum_{b\prime=1}^{B} exp(-\beta_M D(X_i, Y_{b\prime}))}, \tag{4.10}$$

and (iii) the object prototypes are calculated:

$$Y_{b(k)} = \frac{\sum_{i=1}^{I} M_{ib} \sum_{j=1}^{J} m_{ib(jk)} X_{i(j)}}{\sum_{i=1}^{I} M_{ib} \sum_{j=1}^{J} m_{ib(jk)}}. \tag{4.11}$$

Finally, the control parameters β_m and β_M are increased and the loop repeats.

4.1.3 Results

The following results were computed on the 633 sounds classified in the 10 sound classes. The digitalized sounds were spike coded beforehand using 256 spikes for each sound. In any computation step were the distance between two sounds was calculated the amplitude, time and frequency values of the spikes were normalized across both sounds, i.e. the feature space of all spikes was a 3-dimensional unit hypercube.

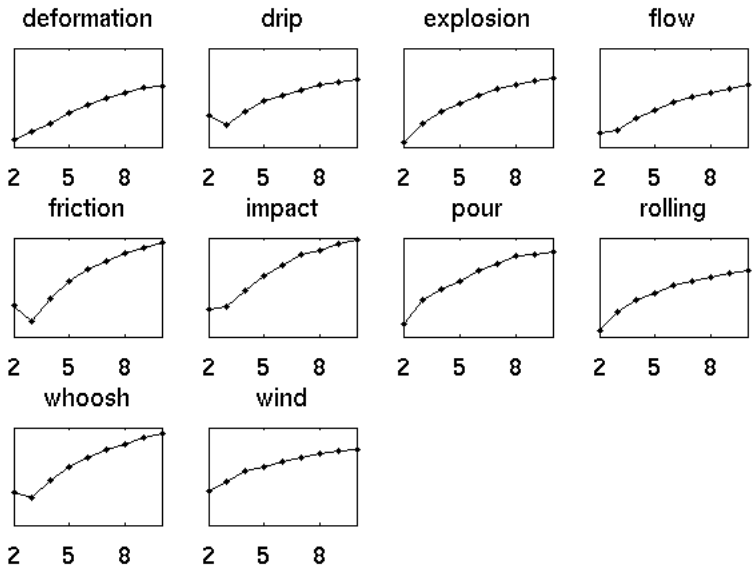


Figure 4.2: Clustering cost (arbitrary units) of sound classes for different numbers of clusters.

Sub-Classes of Sounds

After termination of the pairwise clustering algorithm, the objective function (4.2) yields the costs for that particular partitioning of the data. However, by definition the clustering costs for the same data set decreases monotonically with increasing number of clusters. This fact makes it impossible to use direct comparisons of clustering costs for different numbers of clusters in order to determine the optimal cluster number. we therefore chose to normalize the clustering costs with the number of clusters. Figure 4.2 shows an overview of those normalized clustering cost of all 10 classes of everyday sounds. As can be inferred from the plots, the optimal number of sub-classes is 2 for all classes except for 'Drip', 'Friction' and 'Whoosh'. Those three classes show lower partitioning costs for 3 sub-classes. Since the pairwise clustering algorithm determines the assignments of sounds to their respective cluster, these results can be examined by looking at the distance matrices. Figure 4.3 shows the distance matrix of sound class 'wind' including the assignments of sounds to their corresponding cluster. In order to make the clustering results clearer figure 4.4 shows the same distance matrix but now the sounds are ordered such that they form coherent clusters. One can observe that the red cluster consists of sounds which are very similar to each other, i.e. the distances between each other are very small. This homogeneity is not that pronounced in the blue cluster. Figures 4.5 and 4.6 show the distance matrix for the class 'friction', which was partitioned into three clusters. Although in the unsorted matrix no apparent structure can be distinguished (apart from the small group of sounds at the end), the sorted matrix reveals three clear clusters with small inner-cluster distances. On the other hand,

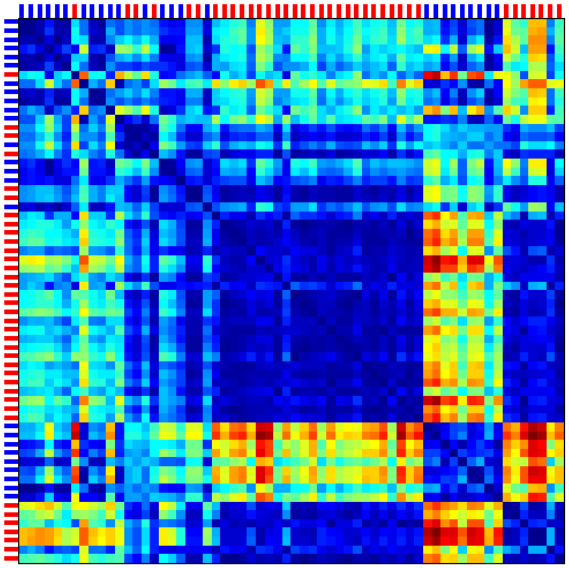


Figure 4.3: Distance matrix of sound class 'wind'. The colored ticks on the axes denote the sound to cluster assignments (ticks of one color constitute one cluster).

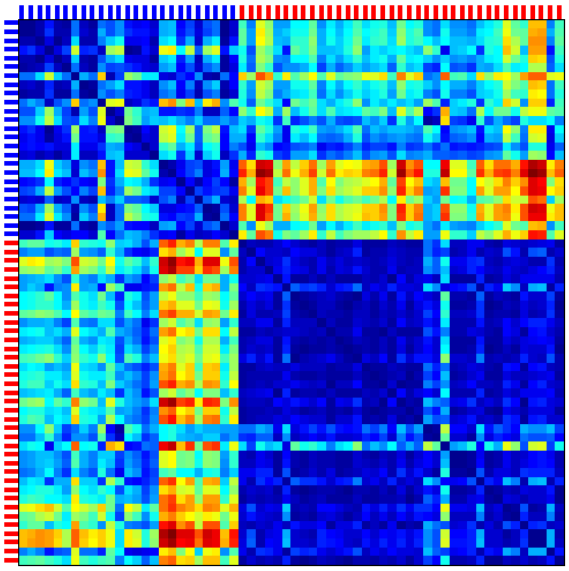


Figure 4.4: Like the distance matrix in figure 4.3 but now the sounds are sorted along the axes forming coherent clusters.

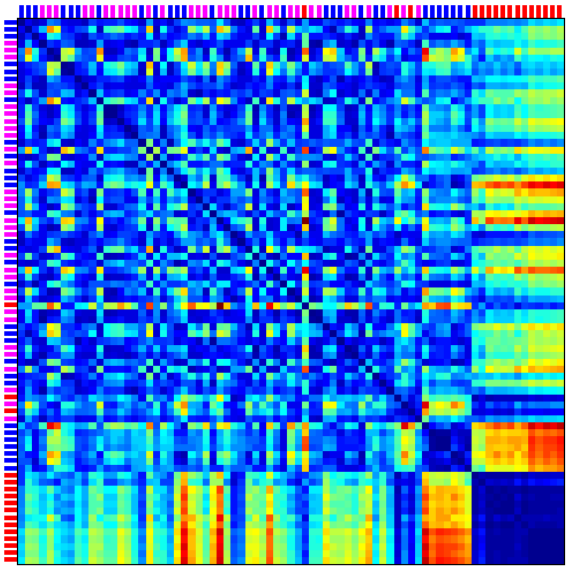


Figure 4.5: Like figure 4.3 but for class 'friction'.

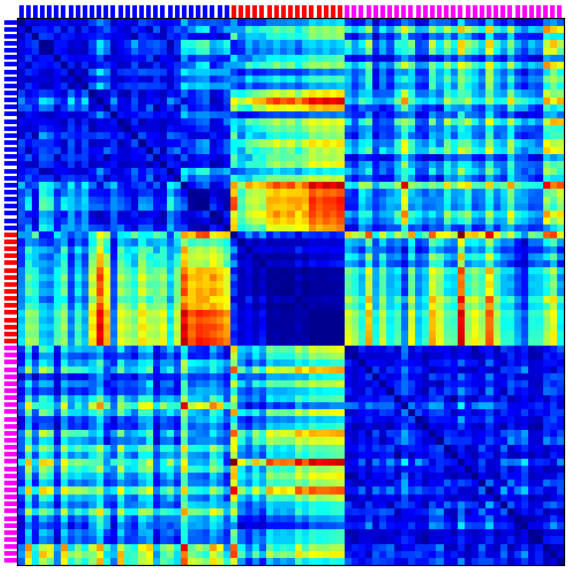


Figure 4.6: Like figure 4.4 but for class 'friction'.

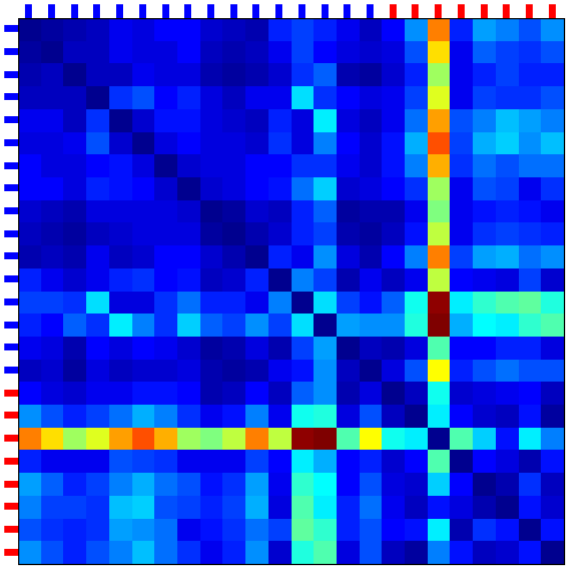


Figure 4.7: Like figure 4.4 but for class 'pour'.

as figure 4.7 shows, the pairwise clustering of the sounds in class 'pour' into two clusters does no reveal two considerably different sub-classes of sounds. Apart from the single clearly different sound all sounds in this class seem to be very alike.

Sound Prototypes

Visual Evaluation

Although the algorithm presented in 4.1.2 includes a clustering phase we did not make use of it, but instead chose to generate the prototypes of the sound clusters determined by the pairwise clustering in 4.1.3. The visual evaluation of the generated prototypes is not really simple, mostly because a prototype sound is the average of multiple sounds, each lying in a 3-dimensional feature space, and thus making it difficult to visually assess the matching of the prototypes to their corresponding member sounds. One possible heuristic way to depict that matching is to compare a prototype with the compound of all its member sound, i.e. showing all spikes of the member sounds as one sound. This is exemplarily shown for sound category 'whoosh' in figures 4.8 and 4.9, which depicts an fairly good matching of the prototypes with their member sounds. Likewise, the three prototypes generated from the sound class 'friction', shown in figure 4.10, have each their own characteristic distribution of spikes. On the other hand, as shown in figure 4.11, the two prototypes generated for the sound category 'pour' don't differ very much, which is consistent with the observation from the previous section that the sounds

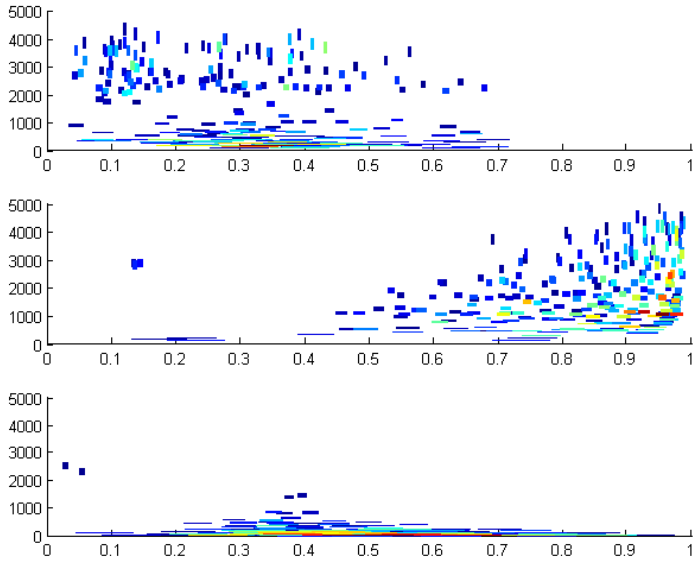


Figure 4.8: Three prototypes generated from the sound class 'whoosh', each with 256 spikes.

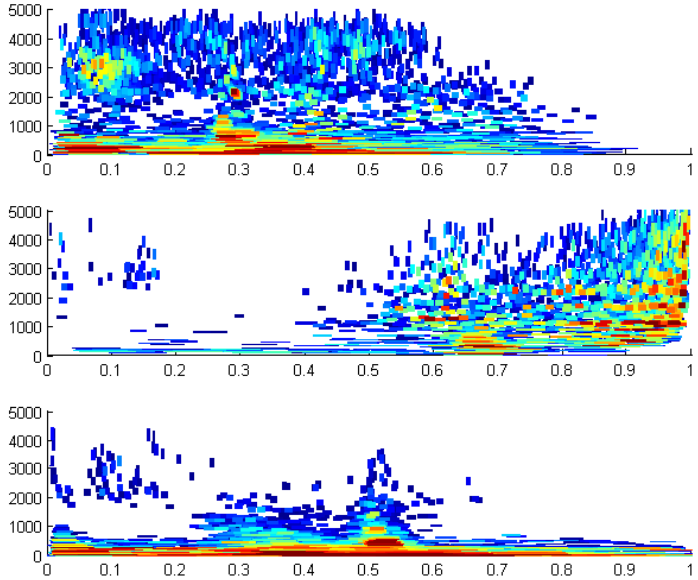


Figure 4.9: Compound sounds from category 'whoosh'. All sounds belonging to one prototype are plotted on top of each other.

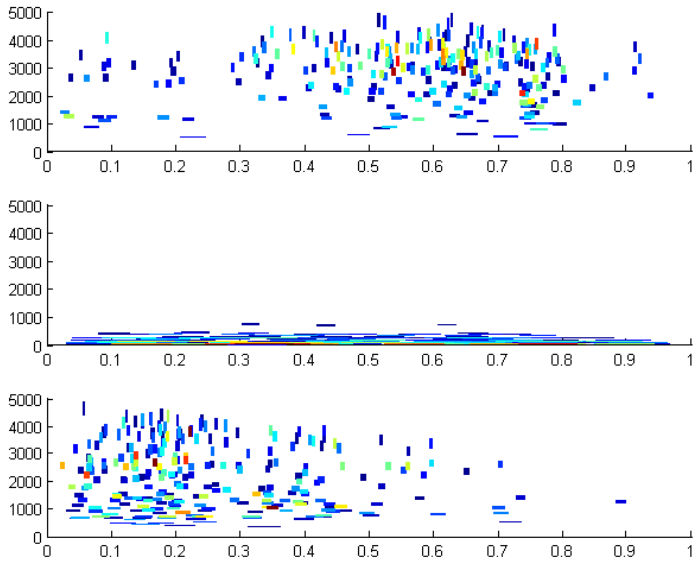


Figure 4.10: Three prototypes generated from the sound class 'friction', each with 256 spikes.

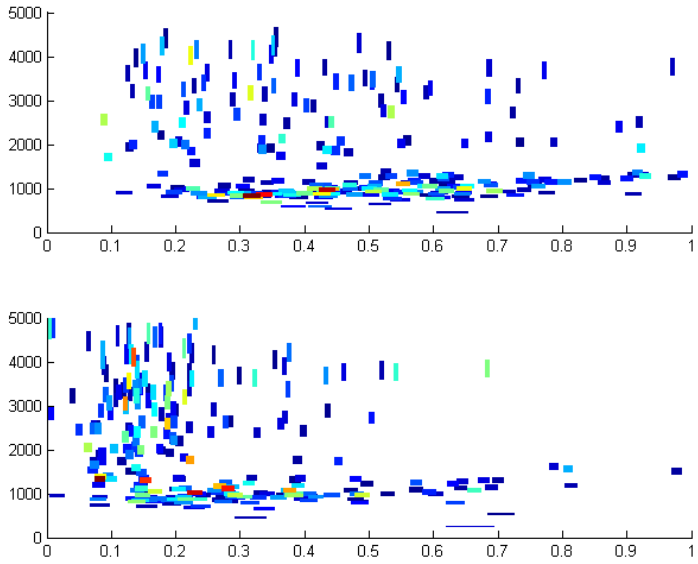


Figure 4.11: Two prototypes generated from the sound class 'pour', each with 256 spikes.

in this category are very similar.

Auditory Evaluation

Experience from previous studies shows that in order to encode a sound with spike coding without substantial loss of quality a minimum of 1000 spikes per second is recommended. Thus, also the auditory evaluation of resynthesized prototypes is not easy since each prototype consists of 256 spikes. This results in resynthesized sounds of rather low quality with a clearly noticeable 'drippiness' of the sound,something typical for spike coded sounds with small number of spikes. Nevertheless, the number of spikes is enough in order to generate sounds that provide a rough intuition about the 'correctness' of the sounds in terms of its membership to a certain category.

4.1.4 Discussion

Both the question of how many sub-classes there are in each sound category and the partitioning of the categories are not trivial. In the approach we chose, each sound category is clustered into groups of sounds by means of pairwise comparisons and thereby maximizing the similarity of sounds in one group. In order to find our the optimal number of clusters all ten categories were partitioned with different numbers of clusters, the resulting clustering costs normalized by the number of clusters and the minimum cost chosen as the optimum. It has to be questioned whether this choice is reasonable in the first place. First of all, there is no formal justification for the normalization of the costs. Furthermore, the possibility is not considered that a category is not partitioned at all. A good example is the case of the pairwise clustering of sound category 'pour' (Figure 4.7). The homogeneity in this category is so obvious that seemingly the optimal choice would be not to partition at all, which is also depicted by the two very similar prototypes (Figure 4.11). An alternative approach in this regard would be to use *hierarchical clustering* on each category and then specify some optimality criterion on the hierarchy in order to define the number of clusters.
Another controversial issue to consider in my approach arises from the fact that the initial classification of the sounds into the ten categories was done supervised. As a consequence one category can contain several sounds which are very similar in the context of perception, whereas the distances of the spike coded sounds is very large. Not only does this lead to unprecise clusterings into sub-categories. Also the computation of reasonable prototypes is made more difficult. For instance, all sounds in the sub-class that contains the sound from class 'drip' in figure 4.12 also consist of temporally disjoined spike volleys each representing the sound of a drop. However, each sound contains a varying number of such drop elements, distributed differently in time. Thus, the averaging process that generates the prototypes destroys the fine temporal alignment of the spike volleys and smears them along the time axis. This is substantiated by looking at the respective prototype of those sounds, shown in the top sound of figure 4.12, were a temporal structure of discrete spike volleys is barely identifiable. One possible solution to this critical point is to cut the original sound signals in a way that they only contain

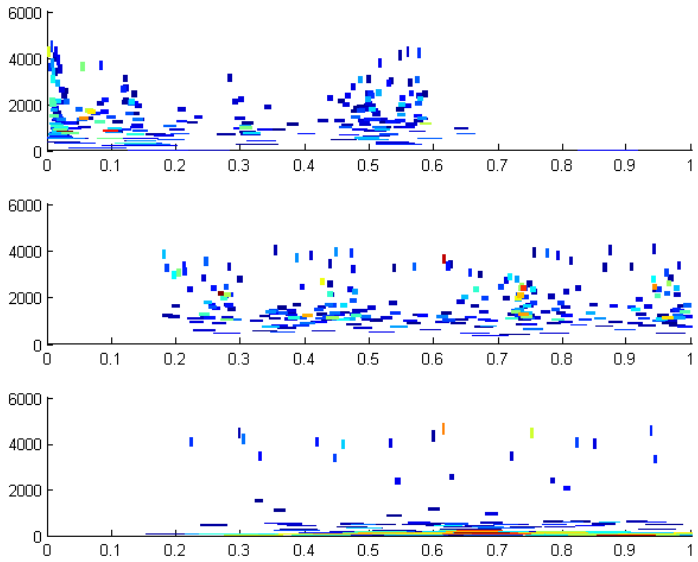


Figure 4.12: Three prototypes generated from the sound class 'drip', each with 256 spikes.

one occurrence of a certain sound event, such as the sound of a drop.

Nevertheless, by individually comparing the graphical descriptions of sounds with the corresponding graphical descriptions of the prototypes, one can state that most of the generated prototypes are reasonably correct average representations of their member sounds. One issue to be considered concerns the normalization of the spike features. This normalization is needed for a precise calculation of sound-to-sound distances. For instance, two sounds that only differ in loudness but are otherwise identical would have nonetheless a big distance without normalizing, whereas by normalizing the amplitudes that distance becomes minimal. Thus, normalization of both amplitudes and frequencies is by and large an appropriate approach. In terms of the time dimension, however, the situation becomes rather ambiguous when normalizing. For instance, when two sounds with the same periodical content but different durations are normalized, the temporal structure of the spikes that describes the periodicity will differ in both after encoding, thus creating an undesired bigger distance. Hence, it would be probably appropriate to use normalized spikes only when calculating distances and to use unnormalized spikes when estimating the sound prototypes.

Having considered all the aforementioned issues, the sound prototypes provided by this study can be used in order to optimize physical models for sound synthesis. Another possible extension to this study is to adjust the weights $c_{a,f,t}$ in the spike-to-spike distance for each sound category individually such that classification results for each category are improved. This can be done by modifying the objective functions presented in section 2 accordingly.

4.1.5 Annotations on Matlab Functions

The Matlab code can be found in the software supplementaries in the directory `audio_prototypes`. See the file `README` for installation instructions.

pwcluster.m

This is the implementation of the pairwise clustering algorithm presented in the methods section. The default values of the optional input arguments have proven to be reasonable choices, although manual adjustments can be taken.

Auxiliary Functions:

- `pwcluster_cost.m` calculates the clustering cost.
- `pwcluster_meanFields.m` calculates the mean fields.
- `pwcluster_monitor.m` plots the development of the clustering costs as the algorithm progresses.

proto.m

This function implements the combined algorithm, as presented in the methods section, which calculates the object prototypes. For defining the correspondence matrix between two objects it can be chosen between the Munkres and the Softassign algorithm. Again, the default values of the optional input arguments have proven to be reasonable choices, although manual adjustments can be made. Note that choosing an α-value other than zero can lead to numerical instability of the algorithm as the annealing process progresses, the reason being that infinite values are produced. In this case the algorithm terminates, which for large number of elements per object can happen long before the convergence criterion has been reached.

Auxiliary Functions:

- `proto_degreeOfConvergence.m` computes the degree of convergence of the correspondence and assignment variables.
- `proto_initPrototypes.m` initializes the prototypes randomly by sampling the feature space of the input data.
- `proto_monitoring.m` plots the development of the degree of convergence as the algorithm progresses.
- `proto_munkres.m` computes the correspondence variables between two objects using the Munkres algorithm.
  Note: Here the Munkres algorithm is an external implementation by Markus Buehren, given in `assignmentoptimal.*`.

- `proto_softassign.m` computes the correspondence variables between two objects using the Softassign algorithm.
- `proto_updatePrototypes.m` updates the prototypes at the end of each iteration.

4.2 Fisher LDA projections

4.2.1 Dimensionality Reduction and Sound control

Fisher's Discriminant [21] (or linear discriminant analysis LDA) was applied to the problem of dimensionality reduction to the data on impact sound classification acquired in Deliverable 5.1 [2], chapter 4. The parameter space that defines the impact models output was reduced to 2 dimensions. This projection can be reversed, such that we can construct a controller for MAX/MSP that is much easier to use the the original 10 sliders for the impact model.

The demonstration system projects first the available data of a chosen subject. The resulting image is then used as a visualization of the parameter space. On top of that a MAX/MSP 2-D controller (slider) is used to change the models parameters using the projected plane. The underlying image is a orientation what regions the subject on which decisions the projection is based. This can be an orientation, but can be left out completely to let the designer explore the plane on his own, without being influenced.

4.2.2 The Projection

Binary case

Input vectors $\mathbf{x} \in \mathbb{R}^n$ get projected down to a one dimensional $y \in \mathbb{R}$ using:

$$y = \mathbf{w}^T \mathbf{x}$$

The formulation of the Fisher criterion for this case having 2 classes C_1 and C_2 with mean vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ is as follows:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

, where

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

is the between-class covariance matrix and

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

is the within-class covariance matrix. Maximizing $J(\mathbf{w})$ finds the best direction to project the data on a line that they are best spearable.

Multiple classes

The multiclass case requires the optimization of:

$$J(\mathbf{W}) = Tr\{(\mathbf{W}\mathbf{S}_W\mathbf{W}^T)^{-1}(\mathbf{W}\mathbf{S}_B\mathbf{W}^T)\}$$

where the weight vectors are arranged in the matrix columns of $\mathbf{W}$. For details refer to [24]. $\mathbf{W}$ itself is a projecting matrix of dimension $n \times d$, that projects n dimensional vectors $\mathbf{x} \in \mathbb{R}^n$ to a d dimensional vector $\mathbf{y} \in \mathbb{R}^d$. (the bias was omitted here).

Before we project a datapoint in practice we need to normalize the parameters to lie in a $[0, 1]$ interval, where frequency and amplitude parameters should be treated logarithmically. This assures an audibly equidistant projection plane.

Given classified data points $\mathbf{X} \in \mathbb{R}^{N \times D}$ and class lables $T \in \{1, 2, 3, 4\}^N$ and the indexes of logarithmical and linear parameters I_{lin}, I_{log} we can compute the projection as follows:

Let $\mathbf{x}_j$ be the j-th column of $\mathbf{X}$ and x_{ij} an element at row i and column j of $\mathbf{X}$.

1. Normalize data:

$$\forall j \in I_{lin} : \tilde{x}_{ij} = \frac{x_{ij} - \min \mathbf{x}_j}{\max(x_{ij} - \min \mathbf{x}_j)} = \frac{x_{ij} - \gamma_j}{\lambda_j}$$

$$\forall j \in I_{log} : \tilde{x}_{ij} = \frac{\ln x_{ij} - \ln(\min \mathbf{x}_j)}{\max(\ln x_{ij} - \ln(\min \mathbf{x}_j))} = \frac{\ln x_{ij} - \gamma_j}{\lambda_j}$$

2. LDA:

$$\mathbf{W}, \mathbf{b} = \underset{\mathbf{W},\mathbf{b}}{\operatorname{argmax}}(J(\mathbf{W}, \mathbf{b}, \tilde{\mathbf{X}}))$$

3. projection (of the whole dataset):

$$\mathbf{Y} = \mathbf{W}\tilde{\mathbf{X}} - \mathbf{b}$$

4. inverse projection (of one selected point using pseudo inverse $\mathbf{W}^\dagger$):

$$\tilde{\mathbf{x}}' = \mathbf{W}^\dagger\tilde{\mathbf{y}} + \mathbf{b}$$

5. denormalisation:

$$\forall j \in I_{lin} : x'_{ij} = \tilde{x}'_{ij}\lambda_j + \gamma_j$$

$$\forall j \in I_{log} : x'_{ij} = \exp\{\tilde{x}'_{ij}\lambda_j + \gamma_j\}$$

x' is the inverse projected parameter vector that is the input for the sound model. For the optimization of the Fisher criterion (2.) we used the implementation of then "dimension reduction toolbox" [73].The projected dataset $\mathbf{Y}$ can be used to plot the classified examples on 2-D plane as depicted in figure 4.14. Any point of the plane (including the black regions) can be projected back into the original space.

By doing so we make a reconstruction error (squared): $MSE = \sum(\mathbf{x}' - \mathbf{x})^2$ which means even when a position at a data point is selected, the sound parameters will not be exactly the same as before the projection. If the mean squared reconstruction error is too large (larger than what?), we recommend to reduce the dimensionality of the problem by ruling out parameters by hand.

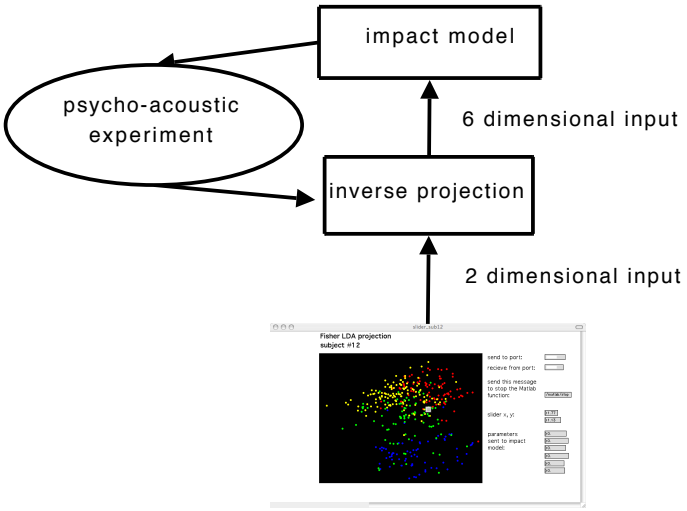


Figure 4.13: Controlling the impact model

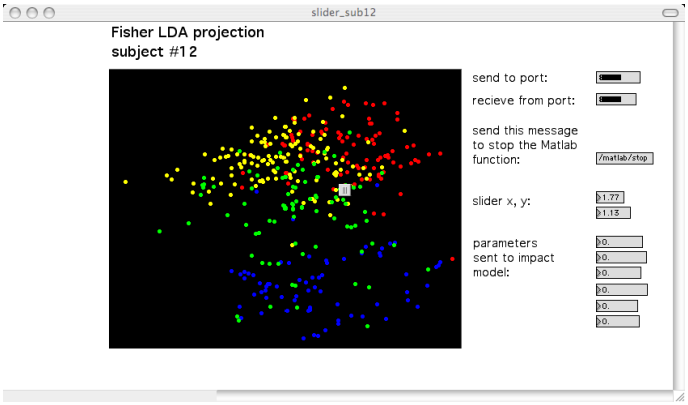


Figure 4.14: Subject 12

4.2.3 Demonstrator in Matlab and MAX/MSP

In figures 4.13 and 4.14 the demonstrator of the projection slider is shown. After classification data from a subject is collected a projection of the data points is done to visualize the data parameter space. One can find 4 regions of the 4 materials according to the subjects answers. By projecting back a 2-dimensional input from the x-y-slider the impact model is now controllable with it, instead of using the normal 6 parameters.

The demonstrator shows that psychophysical experiments can generate data the can be directly be used in a sound design tool. The designer can choose here different points of view, depending on the subject or groups of different subjects or just all.

The Matlab code and MAX/MSP patches can be found in the software supplementaries in the directory `projection_slider`. See the file `README` for installation instructions.y

5 Measurement Assisted Design

5.1 Introduction

Previous research on product-sound quality evaluation [6] shows that sound quality can be described as the *adequacy of a sound attached to the product*. This measure is the combinatory perception of different qualities. In the Bauhaus' notion, basic design elements are always linked to their dynamics and only perceived together (*line is a cause of a moving point.* [43]). We follow this principle in product design looking at sound, object and interaction, which are examined by their combined aesthetics.

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 in a parameter mapping task [41]. In this respect, physically based sound design offers a novel alternative to recording sounds [61]. We measure adequacy of the generated sounds either according to a reference or through judgments of subjects interacting with the object, and optimize the quality of the sounds using statistical methods iteratively.

Therefore finding the optimal sound using a certain sound model is mostly an interactive trial-error process. In this trial-error process, interacting directly with the sound model or indirectly with an artefact, which interacts with the sound model is possible. An interaction with an artefact makes the sound design process more dynamic, because each user interaction, which changes the input parameters of the model, is different from the others. Hence the emerging sound is different.

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 the adequacy of the generated sound in comparison to a given reference or depending on the preference judgments of the users.

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.

5.2 Active Learning Demo

In Deliverable D5.1 [[2]], chapter 4, Active Learning was proposed to guide a search in parameter space of the physical sound model for impact sounds.

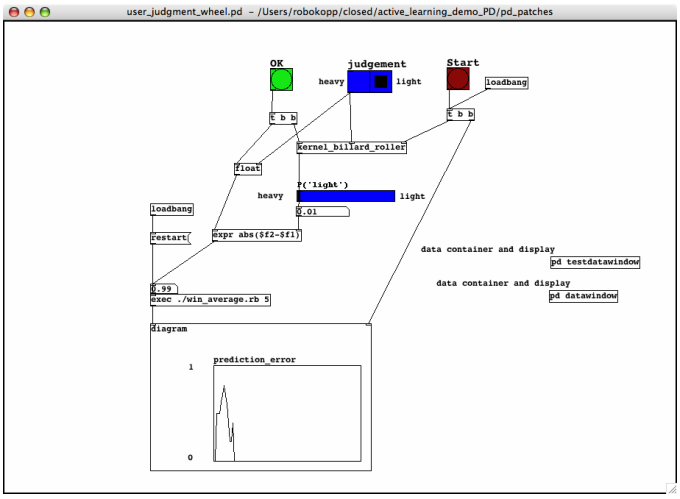
To visualize this process of finding suitable parameter sets we implemented the Active Learning Demo. It uses a *pd* patch (patches) to collect user input and visualize the learning process (Figures 5.1). This kind of a classifier, that is trained interactively here, was proposed in Section 2.2.3 in the Deliverable 5.1 [2].

The examples uses the "rolling" model of the SDT. The subject is asked to specify

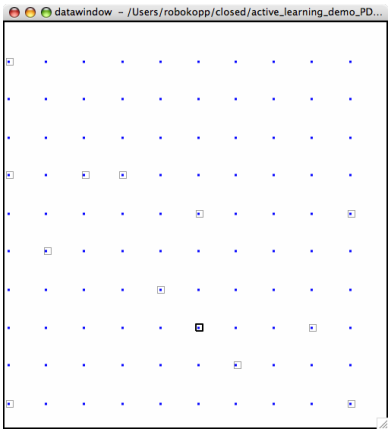
for a given sound, whether it was caused by a light or heavy object. After a couple of labels acquired the classifier finds a decision boundary and tries to fine tune it by picking examples which are expected to carry the most information about the real boundary. To achieve this it minimizes the "Version Space" of the problem, which is the set of all classifiers that are consistent with the so far collected data. Details on the underlying theory of version spaces and learning can be found in [1].

The Demo can be used in 2 modes: 2D and 5D. The latter is to demonstrate higher dimensional problems, which are relevant in practical use. The visualization shows only 2 dimensions of the classifier as well as of the datapool. The 2D mode is used to demonstrate how the decision boundary is found. Here the problem can be visualized completely.

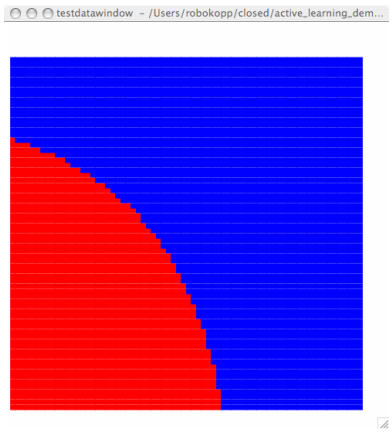
The supplementary software package contains the required patches for *pd* as well as the code for active learning. For Installation please follow the Instructions in the README File.



(a) Active Learning Demo: The PD-Patch controls the active learning program. It plays subsequently sound examples to be labled by the subject, while it is using the labels for training a classifier.



(b) Datapool: gray squares show labeled examples, the black square shows the current example to be labeled.



(c) Classifier: The blue and red regions show the current prediction output. Here it is a linear decision boundary (it looks curved, because the axes are log-scaled). Besides linear boundaries, polynomial and radial basis kernels can be used, with which real curved boundaries and circular regions are possible to learn. However, the more complex the model the higher the danger of overfitting. The choice of decision model has to be made based on the problem.

Figure 5.1: Training and Prediction

5.3 Least Squares in an Auditory Experimental Setup

As mentioned earlier, in the Bauhaus' notion, basic design elements are always linked to their dynamics and only perceived together. We investigate this by using prototypical design elements combining the sound, the object and the interaction: 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 5.2).

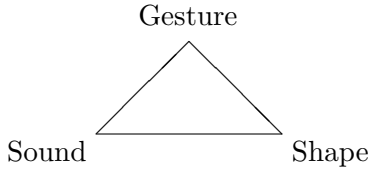


Figure 5.2: threefold aesthetics

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 [22]. 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) [72]. 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 Adaptive Bottle experiment is basically a parameter tuning task, where the preferences of human subjects are mapped to the physically based sound design domain in an interactive environment. A human subject interacts with the bottle, and evaluates the quality of the emerging sound. Based on this evaluation, the input parameters of the physical model are optimized to satisfy the subjects expectations.

5.3.1 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 [53], 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

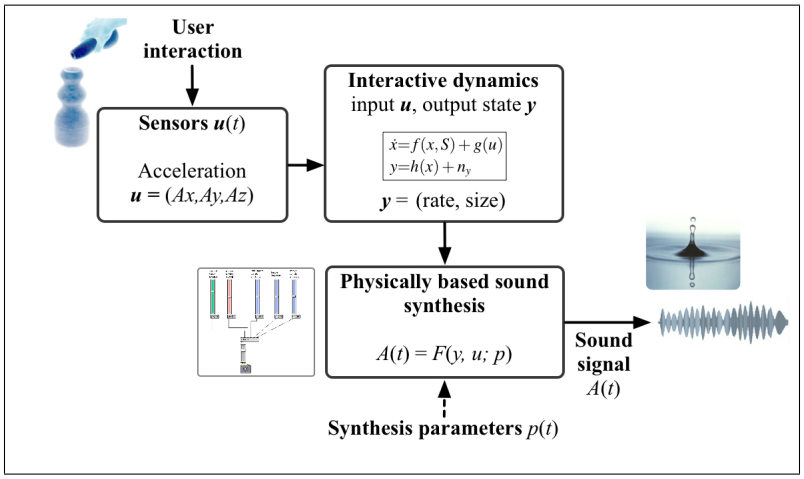


Figure 5.3: The interaction between the subject and the Adaptive Bottle is shown. The acceleration parameters are used for calculating the new volume of the liquid remaining in the bottle, the formation rate and the size of the bubbles. The formation rate and the size of the bubbles are combined with the other parameters to generate the sounds.

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 5.3 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.

5.3.2 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 is left out, because it has been in the Deliverable 5.1 [2]. For details, please refer to the Deliverable 5.1.

5.3.3 The Auditory 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 5.4). 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 $[0, 1]$.

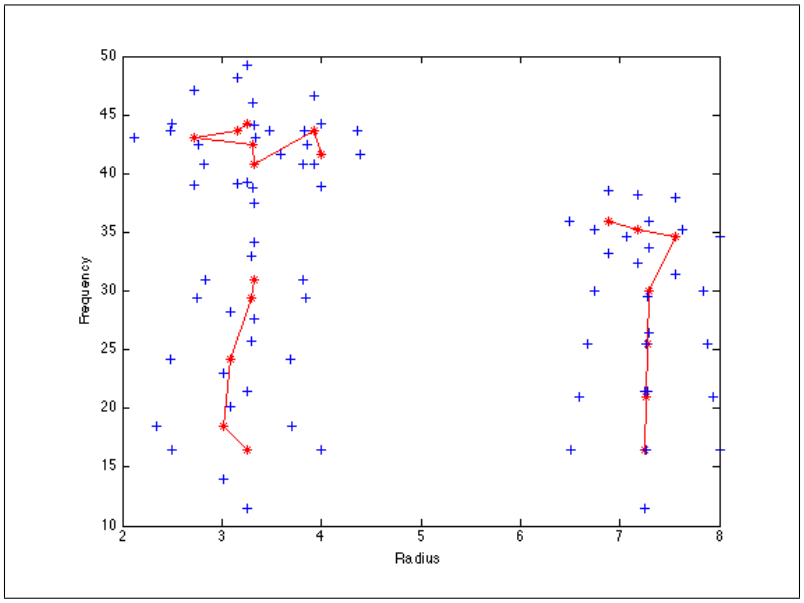


Figure 5.4: The results of one Auditory Adaptive Bottle experiment are shown. The curves and the dots are the learning curves (learning steps) and the data points. The plus signs are the four points calculated around the current data point.

The evaluation process is as follows: 4 parameter settings are available for judgment.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 5.5.

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

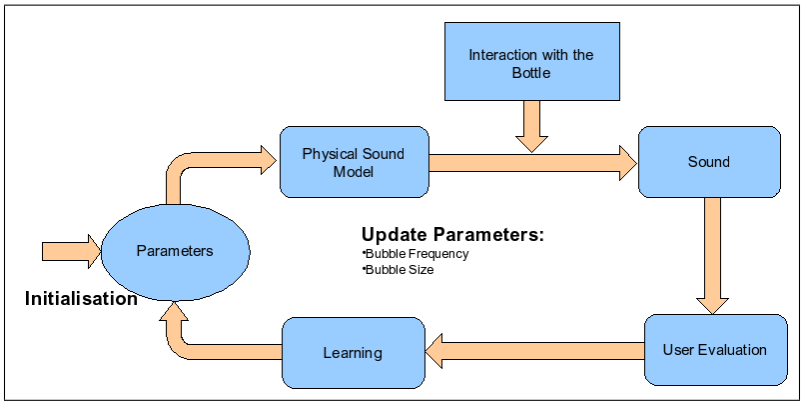


Figure 5.5: The complete scenario of the Adaptive Bottle experiment is shown.

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.

5.3.4 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.

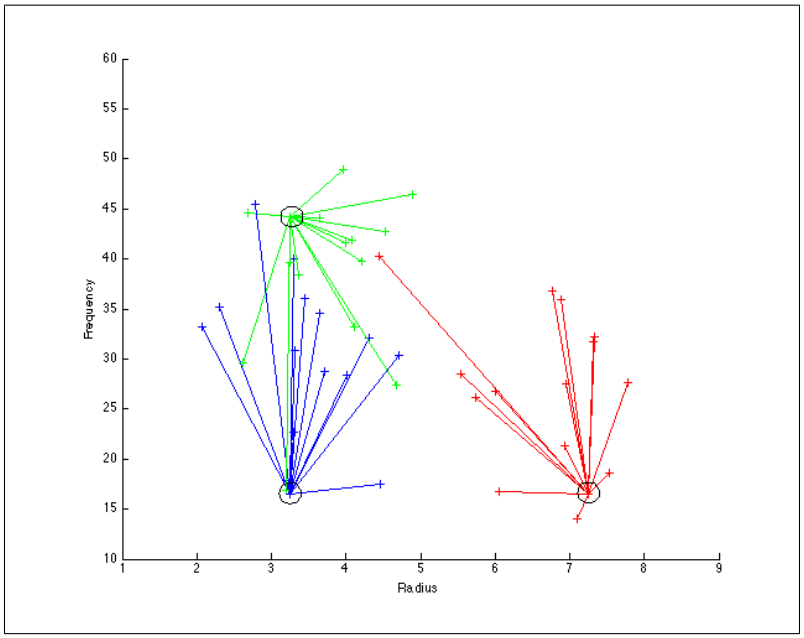


Figure 5.6: The summarized results are shown. The circles indicate the initial points.

The experiments were performed by 15 subjects in total. The summarized results

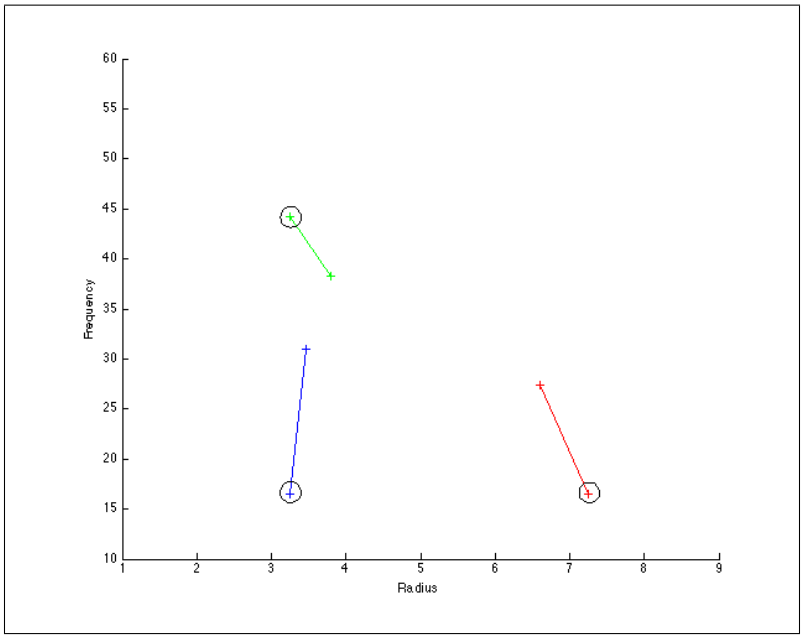


Figure 5.7: The mean of the results are shown. The circles indicate the initial points.

shown in Figure 5.6 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 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 5.7. 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 5.6.

As it can also be seen on Figure 5.6 and 5.7, 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.

5.4 Global Optimization

The least squares method works sufficiently well for a two-dimensional optimization problem in an interactive scenario. However, most of the real world problems are high-dimensional. Similarly a physical sound model has much more input parameters than just two. Therefore a least squares method is not a proper approach for optimizing a complete physical model.

Furthermore, the differences within the physical parameter domain are not perceived monotonically in the perceptual domain. For this reason, a mapping between these two domains must be established. However the perceptual objective function is not known. Fortunately, in both scenarios, namely comparison to a reference and user preferences like in the adaptive bottle scenario the observations can be collected, which in turn can be used to approximate this unknown perceptual objective function.

It is not necessary to approximate the complete perceptual objective function. The problem to solve is in both cases (reference, user preferences) not a regression problem, but an optimization problem. Therefore, the goal is to find the optimal physical parameter settings, which yield the closest sound to the reference or the most preferred one.

A Bayesian treatment to such problems has already been successfully applied recently. Rasmussen [60] showed the success of the Gaussian processes in regression and classification scenarios, in particular for functions with several local minima. Chu and Ghahramani [10] adapted the classification based Gaussian processes to the preferences. These studies established the theory for regression, classification and preference learning, however they did not propose how to select a data point cleverly so that the optimization process can be accelerated in terms of time as well as the number of data points needed. Jones et. al. [14] developed a stochastic method to estimate the improvement each new data point yield. According to this criteria a data point with the largest improvement will be evaluated in a following step. A similar idea based on Bayesian approach is proposed by Osborne et. al. [50]. An expected loss function has been defined to estimate the loss by evaluating one more data point in the optimization process. Brucha et. al. [19] combined the preference learning approach of Chu and Ghahramani with the expected improvement function proposed by Jones et. al. to optimize the rendering surface of objects. As in the optimization of a physical model a rendering surface is defined by a multitude of parameters. Instead of trial and error paradigm Brucha et. al. present the user two new parameter settings, which the user should prefer one or the other. According to the user preferences the system learns perceptual objective function and presents a new pair of parameter settings. The model can learn the perceptual objective function in a feasible amount of time in a six-dimensional parameter space. This result is promising from the physical sound model optimization point of view not only in terms of high dimensionality but also in terms of number of data points needed.

In the following, we introduce the Gaussian process regression and the expected loss criterion to be applied for solving the optimization problem of a physical sound model with respect to a given reference.

5.4.1 Gaussian Process Regression

Gaussian processes offer a powerful method to perform Bayesian inference about unknown functions. A Gaussian process is a collection of random variables, which have a joint Gaussian distribution. A Gaussian distribution is defined over vectors, whereas a Gaussian process is defined over functions. Such a process is fully identified by its mean and covariance functions.

$$f \sim G(\mu, \kappa)$$

meaning that a function f is distributed as a Gaussian process G with the mean function μ and the covariance function κ.

For a Gaussian process an input data point x is associated to a random variable $f(x)$, which is the value of the stochastic function f at that position.

In a Bayesian inference scenario the Gaussian process plays the rule of a prior distribution. A prior distribution does not depend on the observations made about the stochastic function f. Therefore, in the Bayesian treatment, the prior distribution is updated each time an new data has been observed. The distribution in the light of observed data is called the posterior distribution.

The primary goal of computing the posterior distribution is to make predictions for unseen data points. Suppose that $\mathbf{f}$ is the vector of observed function values for the input data points $\mathbf{X}$. Assuming that this data is normally distributed, the unseen data points $\mathbf{X}_*$ and their corresponding function values $\mathbf{f}_*$ represent a joint Gaussian distribution:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = N\left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma} & \mathbf{\Sigma}_* \\ {\mathbf{\Sigma}_*}^T & \mathbf{\Sigma}_{**} \end{bmatrix} \right),$$

where μ represents the mean values of the observed data points and μ the mean values of the unseen data points. Similarly, $\mathbf{\Sigma}$ indicates the covariances for the observed data points, $\mathbf{\Sigma}_*$ indicates the covariances between the observed and unseen data points, and $\mathbf{\Sigma}_{**}$ indicates the covariances for the unseen data points.

Since we know the function values of the observed data points, we are interested in the function values of the unseen data points, which can be calculated by the conditional distribution of $\mathbf{f}_*$ given $\mathbf{f}$:

$$\mathbf{f}_* | \mathbf{f} \sim N(\mu_* + {\mathbf{\Sigma}_*}^T \mathbf{\Sigma}^{-1}(\mathbf{f} - \mu), \mathbf{\Sigma}_{**} - \mathbf{\Sigma}_*^{\mathbf{T}} \mathbf{\Sigma}^{\mathbf{-1}} \mathbf{\Sigma}_*),$$

where the first term of the distribution on the right hand-side of the equation is the posterior mean and the second term is the posterior covariance. The posterior covariance is simply the prior covariance minus a positive term, which means that the more data points are observed, the smaller the posterior covariance becomes.

Equipped with the Gaussian process regression, we need to cleverly select the next data point to be evaluated. For this we utilize the expected loss function.

5.4.2 Expected Loss Function

A Gaussian process regression model can predict the function values of the unseen data points by making use of the information gained over the observed data points. In stan-

dard regression problems the unseen data points are selected randomly. However, in particular, for the optimization of a physical sound model according to used preferences, we need to find the optimum with as less number of trials as possible, because a user subject is not able to evaluate much more than 100 data points. Therefore these data points must be selected cleverly according to a criterion, which takes the gain into account.

The expected loss function considers the loss (gain) by evaluating one more data point before stopping the optimization process. Suppose that we evaluated the datapoints $(\mathbf{x_0}, \mathbf{y_0})$, where $y = \mathbf{f}(x)$. The optimum value obtained from these data points is defined as

$$\eta = \min \mathbf{y_0}.$$

Hence, the loss of evaluating one last data point (x_n, y_n) is defined as

$$\lambda(y_n) = \begin{cases} y_n; & y_n < \eta \\ \eta; & y_n > \eta \end{cases}$$

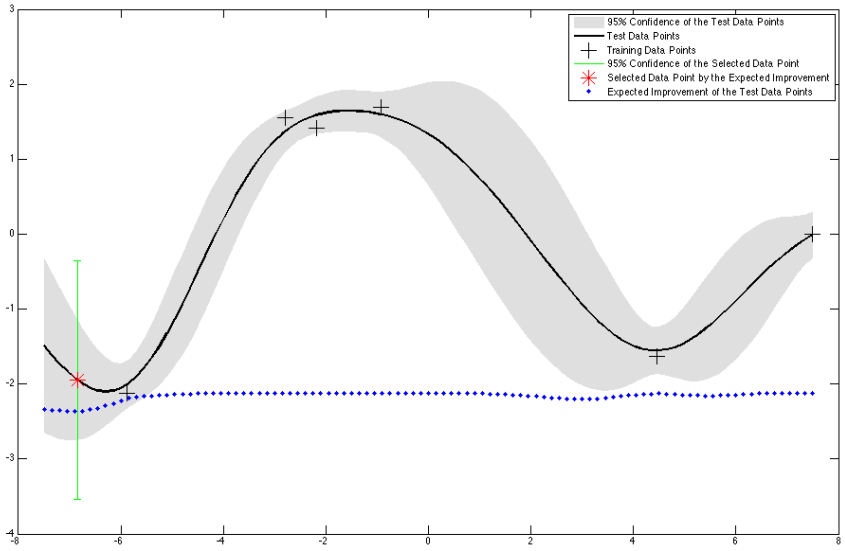
Given the Gaussian process over $\mathbf{f}$, the total expected loss by selecting a data point (x, y) is defined as

$$\begin{aligned} \Lambda(x) &= \int \lambda(y) p(y|x) dy, \\ &= \eta + (\mu - \eta)\Phi(\eta, \mu, \Sigma) - \Sigma N(\eta, \mu, \Sigma). \end{aligned}$$

In this formulation, μ and Σ represent the posterior mean and covariance functions respectively. N is the probability distribution function for the Gaussian distribution, and Φ is the cumulative distribution function for the same distribution. The location of the data point x, where the expected loss function is the lowest indicate the optimal location for the next function evaluation. Note that $\Lambda(x)$ decreases, when the mean μ becomes lower than η, and when the covariance Σ becomes larger. The first case indicates an exploitation effect, and the second case the exploration effect. Having defined the expected loss function, in each step this function should be minimized, in order to obtain the data point to be evaluated in the following step. Since this function is continuous and differentiable in the whole parameter domain, it is possible to find the minimum of the function in each step.

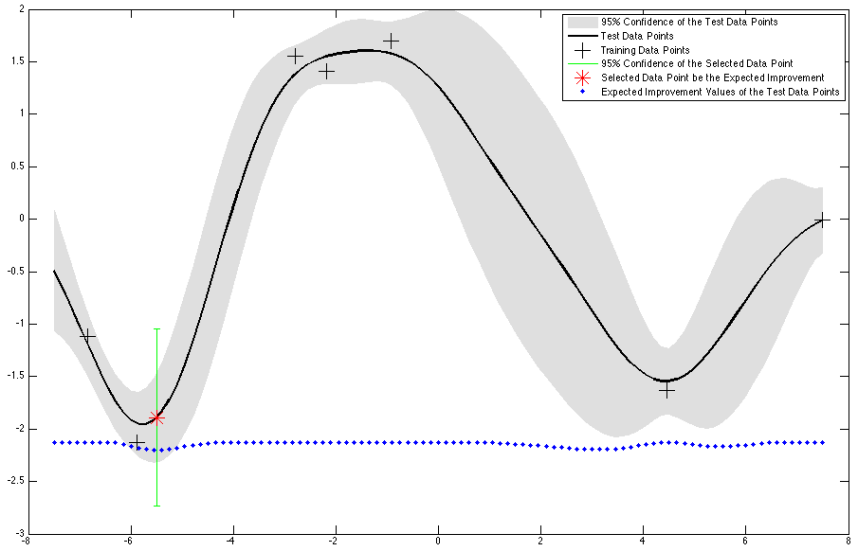
Figures 5.8 5.9 indicate three iterations made for a one dimensional function. In each step, the expected loss function indicates, where the minimum function is expected. From these three iterations, the exploitation and exploration steps can be observed very easily. In the first and third iterations, the function exploits, the region, where the current minimum is located. In the second iteration, however, the function diverges to an unvisited point, where the function $\mathbf{f}$ has not been tested so far.

The expected loss function in combination with the Gaussian processes can easily be applied onto high-dimensional cases. Furthermore, the same approach can be adapted to the preference based experimental setups to select the next data point to be presented to the user subject. Therefore this method is promising for the optimization problems of physical sound models.

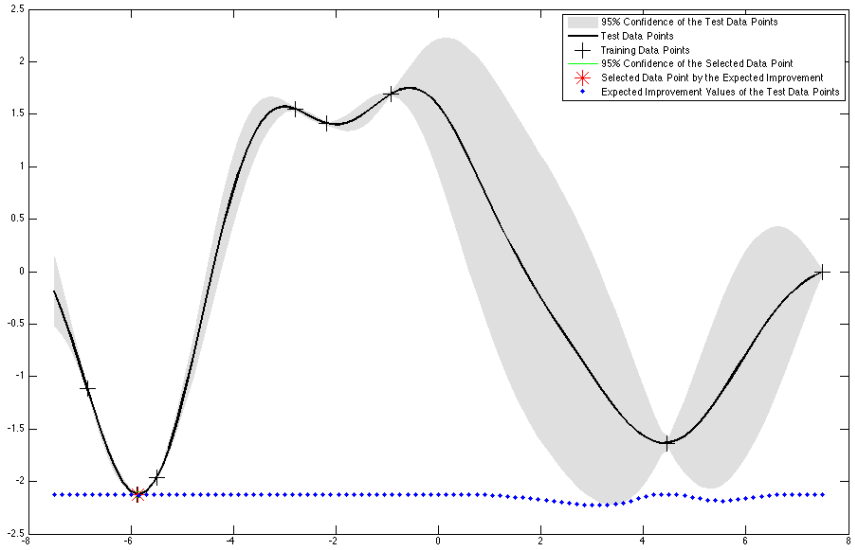


(a) Expected Improvement - Evaluation 3

Figure 5.8: This figure together with Figure 5.9 indicate three iteration steps of the function evaluation and new data point selection, which are performed alternately. The black curve within the shaded area indicates the function values of the test data points. The shaded area indicates the 95% confidence interval of the function values of these data points. The data points indicated by the plus sign are the training data points. The blue dots almost below the shaded area are the expected loss function values of the test data points. Finally, the red asterisk indicates the selected data point for the next iteration.



(a) Evaluation 4



(b) Evaluation 5

Figure 5.9: This figure indicates two following iteration steps following Figure 5.8 of the function evaluation and new data point selection. See Figure 5.8 for detailed explanation.

5.5 Conclusion

In the least squares optimization setup, 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.

However, the least squares solution cannot be adapted to high-dimensional problems due to the exponentially increasing data point evaluations. Therefore, we propose a Bayesian inference based function optimization and data selection method. This method is suitable for high-dimensional problems. It can be incorporated both for reference based and preference based optimization problems.

5.6 Outlook

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, in order to solve the high-dimensional problem, the Bayesian inference model is proposed to use the prior knowledge obtained in the previous iterations to calculate a posterior probability of the direction of the next learning step. Hence, this model will be integrated into the physical sound models and tested in two optimization scenarios. The first scenario will be to use a reference sound to optimize the model. In this scenario, the optimal input parameters will be searched, which generate the closest sound to the given reference sound. In the second scenario, a user subject will be needed to evaluate given data points in a psychoacoustical experiment. In such an experiment, the user subject will be presented two generated sounds, between which he / she should prefer the one or the other sound. According to this decision, the next pair of data points (generated sounds) will be determined. The Gaussian processes combined with the expected loss function is capable of solving both of these problem types.

A Publication List

- K. Adiloglu, R. Annies, K. Obermayer, Y. Visell and C. Drioli, "Adaptive Bottle". Proceedings of the International Computer Music Conference, August 2008, Belfast, Ireland.
- K. Adiloglu, R. Annies, F. Henrich, A. Paus, K. Obermayer, "Geometrical Approaches to Active Learning". Proceedings of the Workshop Autonomous Systems – Self-Organisation, Management, and Control. October 2008. Shanghai, China.
- R. Annies, E. Martinez, K. Adiloglu, H. Purwins, K. Obermayer, "Classification Schemes for Step Sounds Based on Gammatone Filters". Music, Brain and Cognition. Part 2: Models of Sound and Cognition. Neural Information Processing Systems Conference (NIPS). Whistler, Canada. December 7-8, 2007.
- R. Annies, K. Adiloglu, E. Wahlen, H. Purwins, K. Obermayer. "A Graphical Representation and Dissimilarity Measure for Basic Everyday Sound Events". Submitted to the IEEE Transactions on Audio, Speech and Language Processing, 2009.
- Robert Annies, Olivier Houix , Hendrik Purwins, Nicolas Misdariis, Kamil Adiloglu , Antoine Minard, Klaus Obermayer: Timbre Feature Reduction for the Classification of Perceptual Relevant Categories of Everyday Sounds prepared for submission to Speech Communication, 2009.
- Robert Annies, Kamil Adiloglu, Hendrik Purwins, "Playing with SID", submitted to Sound and Music Computing Conference (Special Session), 2009.
- S. de Bruijn, H. Purwins, R. Annies, C. Laurier, K. Adiloglu, K. Obermayer, Comparative Prediction of Emotional Attributes of Sound and Music Bites", Technical Report Berlin Institute of Technology, 2009.
- S. de Bruijn, H. Purwins, R. Annies, C. Laurier, K. Adiloglu, K. Obermayer, " An Emotional Color Palette for the Sound Artist" , Manuscript, 2009.
- C. Drioli, P. Polotti, D. Rocchesso, S. Delle Monache, K. Adiloglu, R. Annies, K. Obermayer, "Auditory Representations as Landmarks in the Sound Design Space", To be published in the Proceedings of Sound and Music Computing Conference. Porto, Portugal, July 2009.
- E. Martinez, K. Adiloglu, R. Annies, H. Purwins, K. Obermayer. "Biologically inspired classification of everyday sounds". Technical Report. 2007

- E. Martinez, K. Adiloglu, R. Annies, H. Purwins, K. Obermayer. "Classification of everyday sounds using perceptual representation". Proceedings of Audio Mostly. 2nd Conference on Interaction with Sound. Ilmenau, Germany. September 27-28, 2007.
- Matthias Schultze-Kraft, "Generating Prototypes of Everyday Sounds", Technical Report, Berlin Institute of Technology, 2008.
- P. Susini, N. Misdariis, G. Lemaitre, D. Rocchesso, P. Polotti, K. Franinovic, Y. Visell, K. Obermayer, H.Purwins, K. Adiloglu,"Closing the Loop of Sound Evaluation and Design". Proceedings of the 2nd ISCA/DEGA Tutorial and Research Workshop on Perceptual Quality of Systems. Berlin, Germany. 2006
- H. Purwins, P. Holonowicz, P. Herrera, (In Press). Prediction of Surprise Based on Loudness - A Preliminary Study . Sound and Music Computing Conference, Porto, 2009.
- Purwins, H. and Hardoon, D. (2009). Trends and Perspectives in Music Cognition Research and Technology. Connection Science. 21(2-3), 85-88.
- Elio Wahlen. "Gehörgerechte Klangrepräsentation für automatische Merkmalsklassifikationen". Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2008 .

Bibliography

[1] K. Adiloğlu, R. Anniés, F. Henrich, A. Paus, and K. Obermayer. Geometrical approaches to active learning. In *Autonomous Systems – Self Organization, Management and Control*, 2008.

[2] K. Adiloğlu, Robert Anniés, Hendrik Purwins, and Klaus Obermayer. Closing the loop of sound evaluation and design (closed), deliverable 5.1: Representations and predictors for everyday sounds. Technical report, Technische Unversität Berlin, 2008.

[3] R. Anniés, E. Martinez Hernandez, K. Adiloğlu, H. Purwins, and K. Obermayer. Classification schemes for step sounds based on gammatone-filters. In *NIPS-Workshop Music, Brain, & Cognition*, 2007.

[4] J. J. Aucouturier, B. Defreville, and F. Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *Journal of the Acoustical Society of America*, 122(2):881–891, 2007.

[5] E. Bigand, S. Vieillard, F. Madurell, J. Marozeau, and A. Dacquet. Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts. *Cognition & Emotion*, 19(8):1113–1139, 2005.

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

[7] M. Casey. Mpeg-7 sound-recognition tools. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):737–747, Jun 2001.

[8] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] S. Chu, S. Narayanan, and C.-C. J. Kuo. Environmental sound recognition using MP-based features. In *Proceedings of ICASSP*, 2008.

[10] W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proceedings of the International Conference on Machine Learning*, 2005.

[11] M. Coath, S. L. Denham, L. M. Smith, H. Honing, A. Hazan, P. Holonowicz, and H. Purwins. Model cortical responses for the detection of perceptual onsets and beat tracking in singing. *Connection Science*, 2009.

[12] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[13] M. Cowling and R. Sitte. Comparison of techniques for environmental sound recognition. *Pattern Recognition Letters*, 24(15):2895 – 2907, 2003.

[14] M. Schonlau D. R. Jones and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13-4:455–492, 1998.

[15] S. B. Davis and P. Mermelstein. Evaluation of acoustic parameters for monosyllabic word identification. *The Journal of the Acoustical Society of America*, 64(S1):S180–S181, 1978.

[16] B. Defréville, P. R., C. Rosin, and F. Pachet. Automatic recognition of urban sound sources. In *120th AES Convention*, 2006.

[17] C. Drioli, P. Polotti, D. Rocchesso, S. Delle Monache, K. Adiloglu, R. Annies, and K. Obermayer. Auditory representations as landmarks in the sound design space. In *Proceedings of Sound and Music Computing Conference*, 2009.

[18] D. Dufournet, P. Jouenne, and A. Rozwadowski. Automatic noise source recognition. *The Journal of the Acoustical Society of America*, 103(5):2950, May 1998.

[19] N. de Freitas E. Brucha and A. Ghoosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

[20] T. Eerola. Ingredients of emotional music: An overview of the features that contribute to emotions in music. In *Frontiers in Human Neuroscience. Conference Abstract: Tuning the Brain for Music*, 2009.

[21] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[22] 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.

[23] A. Friberg, R. Bresin, and J. Sundberg. Overview of the kth rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3):145–161, 2006.

[24] K. Fukunaga. *Introduction to Statistical Pattern recognition*. Academic Press, 1990.

[25] P. Gaunard, C. G. Mubikangiey, C. Couvreur, and V. Fontaine. Automatic classification of environmental noise events by hidden markov models. In *IEEE ICASP*, volume 6, pages 3609–3612, 1998.

[26] W. W. Gaver. How do we hear in the world? explorations of ecological acoustics. *Ecological Psychology*, 5(4):285–313, 1993.

[27] W. W. Gaver. What in the world do we hear? an ecological approach to auditory event perception. *Ecological Psychology*, 5-4:285–313, 1993.

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

[29] S. Gold, A. Rangarajan, C-P. Lu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31:957–964, 1998.

[30] S. Gold, A. Rangarajan, and E. Mjolsness. Learning with preknowledge: clustering with point and graph matching distance measures. *Neural Computation*, 8:787–804, 1996.

[31] R. Gribonval and E. Bacry. Harmonic decomposition of audio signals with matching pursuit. *IEEE Transactions on Signal Processing*, 51(1):101–111, 2003.

[32] E. Guaus and P. Herrera. Music genre categorization in humans and machines. In *Proceedings of the 121st convention of the audio engineering society*, 2006.

[33] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389422, 2002.

[34] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, 1998.

[35] A. Hazan, R. Marxer, P. Brossier, H. Purwins, P. Herrera, and X.Serra. What/when causal expectation modelling applied to audio signals. *Connection Science*, 2009.

[36] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1–14, 1997.

[37] 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.

[38] O. Houix, N. Misdariis G. Lemaitre, P. Susini, and J. Tardieu. Closing the loop of sound evaluation and design (CLOSED): Deliverable 4.3, sonic interactions, naturalness, usability and emotions. premises for a computing model. Technical report, IRCAM (Paris), 2009.

[39] O. Houix, G. Lemaitre, N. Misdariis, and P. Susini. Classification of everyday sounds: Influence of the degree of sound source identification. *The Journal of the Acoustical Society of America*, 123(5):3414, 2008.

[40] Olivier Houix, Guillaume Lemaitre, Nicolas Misdariis, Patrick Susini, Karmen Franinovic, Daniel Hug, Jacqueline Otten, Jill Scott, Yon Visell, Delphine Devallez, Federico Fontana, Stefano Papetti, Pietro Polotti, and Davide Rocchesso. Everyday sound classification. part 1 : State of the art. Technical report, Commission Europenne, 2003.

[41] 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.

[42] P. Juslin and D.Vstfjll. Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and Brain Sciences*, in press.

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

[44] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[45] C. Laurier and P. Herrera. Audio music mood classification using support vector machine. In *Proceedings of the 8th international conference on music information retrieval*, 2007.

[46] C. Laurier and P. Herrera. Automatic detection of emotion in music: Interaction with emotionally sensitive machines. *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*, pages 9–32, in press.

[47] G. Lemaitre, O. Houix, N. Misdariis, and P. Susini. Naïve and expert listeners use different strategies to categorize everyday sounds. *The Journal of the Acoustical Society of America*, 123(5):3689, 2008.

[48] Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *IEEE International Conference on Computer Vision*, 2001.

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

[50] R. Garnett M.A. Osborne and S. J. Roberts. Gaussian processes for global optimization. In *International Conference on Learning and Intelligent Optimization (LION3)*, 2009.

[51] 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.

[52] R. Meddis. Simulation of auditory-neural transduction: Further studies. *J. of the Acoustical Soc. of America*, 83(3):1056–1063, 1988.

[53] 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.

[54] 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.

[55] 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.

[56] Geoffroy Peeters and Emmanuel Deruty. Automatic morphological description of sounds. *The Journal of the Acoustical Society of America*, 123(5), 2008.

[57] Geoffroy Peeters and Xavier Rodet. Automatically selecting signal descriptors for sound classification. In *ICMC 2002*, 2002.

[58] H. Purwins and D. Hardoon. Trends and perspectives in music cognition research and technology. *Connection Science*, 2009.

[59] Hendrik Purwins, Perfecto Herrera, Maarten Grachten, Amaury Hazan, Ricard Marxer, and Xavier Serra. Computational models of music perception and cognition I: The perceptual and cognitive processing chain. *Physics of Life Reviews*, 5:151–168, 2008.

[60] C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

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

[62] J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[63] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43– 49, 1978.

[64] B. Smith. PsiExp: An environment for Psychoacoustics experimentation using IRCAM musical workstation. In *Musical Perception and Cognition Conference 95*, Berkeley, USA, 1995.

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

[66] Ludger Solbach, Rolf Wöhrmann, and Jörg Kliewer. *The complex-valued continuous wavelet transform as a preprocessor for auditory scene analysis.* L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1998.

[67] C. Spevak and R. Polfreman. Sound spotting a frame-based approach. In *ISMIR*, 2001.

[68] P. Susini, N. Misdariis, G. Lemaitre, O. Houix, D. Rocchesso, P. Polotti, K. Franinovic, Y. Visell, K. Obermayer, H. Purwins, and K. Adiloglu. Closing the loop of sound evaluation and design. *Perceptual Quality of Systems*, 2006.

[69] Andrey Temko, Enric Monte, and Climent Nadeu. Comparison of sequence discriminant support vector machines for acoustic event classification. In *Proceedings of ICASSP*, 2006.

[70] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 9(6):2319–2323, 2000.

[71] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Audio, Speech and Language Processing*, 10(5):293–302, 2002.

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

[73] L. J. P. van der Maaten. An introduction to dimensionality reduction using matlab. Technical report, 2007.

[74] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review. *Neurocomputing*, 2008. Submitted.

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

[76] D. Vstfjll, M. Kleiner, and T. Grling. Affective reactions to and preference for combinations of interior aircraft sound and vibration. *The International Journal of Aviation Psychology*, 13(1):33–47, 2003.

[77] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1999.

[78] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems).* Morgan Kaufmann, 2005.