RELEVANCE-BASED INTERACTIVE OPTIMIZATION OF SONIFICATION

Thomas Hermann, Kerstin Bunte, Helge Ritter

Neuroinformatics Group
Faculty of Technology, Bielefeld University
Bielefeld, Germany
thermann@techfak.uni-bielefeld.de

ABSTRACT

This paper presents a novel approach for the *interactive optimization* of sonification parameters. In a closed loop, the system automatically generates modified versions of an initial (or previously selected) sonification via gradient ascend or evolutionary algorithms. The human listener directs the optimization process by providing relevance feedback about the perceptual quality of these propositions. In summary, the scheme allows users to bring in their perceptual capabilities without burdening them with computational tasks. It also allows for continuous update of exploration goals in the course of an exploration task. Finally, *Interactive Optimization* is a promising novel paradigm for solving the mapping problems and for a user-centred design of auditory display. The paper gives a full account on the technique, and demonstrates the optimization at hand of synthetic and real-world data sets.

[Keywords: Sonification, Data Mining, Evolutionary Algorithms, Parameter Mapping Sonification]

1. INTRODUCTION

Sonification, the auditory representation of data as sound, is a particularly attractive approach to investigate high-dimensional data since sound allows easily to express a multitude of different characteristics in a single sonic event, like for instance by using pitch, level, source location and distance, timbre, timbre change, amplitude envelope, etc. as dimensions. A frequently seen (or: heard) sonification technique is *Parameter Mapping Sonification* (PMS), where these attributes are computed by mapping different data features [1, 2, 3]. A more generalized approach is Multidimensional Perceptual Scaling (MPS), where a linear mixture of data vectors is computed to obtain values to be mapped to acoustic features [4].

However, the more elaborate the techniques are to render data displays (be it sonifications or visualizations), the more difficult becomes the task of tuning the display, and/or adjusting the numerous available method parameters. Until now, in most cases these parameters are subject to manual adjustment by the user (who often is the same as the programmer of the sonification). However, the problem here is that those who are able to understand the parameters and their role within the display are eventually not those familiar with the data, or those who are best skilled in discerning structure from the display (e.g. a physician with trained listening skills and domain knowledge might be an excellent listener but lack know-how to adjust parameters in a senseful way).

In this paper we introduce a novel approach to automatically optimize parameters of sonification systems (demonstrated at hand of a Parameter Mapping Sonification technique), so that the system can be operated by the listener alone, without any need of specific programming knowledge or knowledge about the sonification technique. The key idea is to automatically create modified sonifications starting from either an initial parameter guess or the previously selected sonification. These sonifications are presented to the user via a graphical user interface that allows the user to specify their perceptual quality in terms of a *relevance rating*. The system incrementally adjusts an internal relevance map in parameter space which is in the turn of the optimization loop used to filter newly generated examples. This relevance-based guidance towards promising regions in parameter space is complemented by cost function terms for novelty and structural richness. In summary, a closed-loop adaptive interactive sonification system emerges that is very intuitive to navigate. It enables an improved work division between the programmer and the user of the sonification system. A related approach, but applied for subjective HRTF selection was discussed by Runkle et al. [5].

In detail, we have implemented two optimization techniques to navigate the parameter spaces of sonification techniques, the first using Evolutionary Algorithms, and the second using Gradient Ascend in different variants of a quality function. We introduce the technique, and demonstrate its successful application at hand of two data sets. Specifically we will show how clustering structure in data sets from classification problems can be discovered and audible contrast can be maximized by using the technique. We furthermore discuss the use of the technique beyond Parameter Mapping Sonification in the field of exploratory data analysis.

2. RELEVANCE-BASED OPTIMIZATION OF SONIFICATIONS

Relevance-based optimization aims at allowing users to operate a sonification system without any knowledge of the sonification techniques or the available parameters and their proper values. This is achieved via an iterative optimization system where the human user is able to concentrate on this judging task, removing any distractions and other tasks during the interaction. In result, starting either from an initial sonification or from the previously selected sonification, modified sonifications are generated, and the only task given to the user is to rate their quality, their relevance, or their value to learn something about the data.

2.1. System Overview

For such a system to work we need to define various components such as (a) a sonification technique to generate data representations, (b) a parameter space which covers a large set of potential sonifications, (c) a technique to create descendants (or children) sonifications for a given sonification, (d) a knowledge representation collecting all user feedback during the interaction, (e) a user interface that closes the loop. In this section, we describe the developed components with a particular focus on the most critical component (c) which determines how novel descendants are computed using the available knowledge gathered from user feedback. Figure 1 shows an information flow diagram and all components.

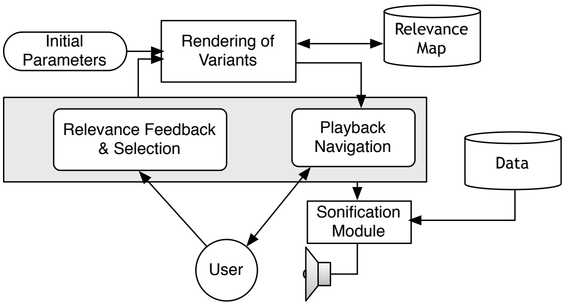


Figure 1: Information Flow Diagram of the Relevance-based Interactive Sonification System.

2.2. Parameter Spaces for the Optimization of Sonifications

Any sonification technique can be formalized as a function $s(X, \theta, T)$ that determines how a data set X and optional excitory actions T will cause the creation of the sonification s. For many techniques like for instance Parameter Mapping Sonification, the action T is only a trigger. For Model-based Sonification [4], T includes the detailed excitory interactions. Typically the sonification can be specified by a parameter vector θ, which for Parameter Mapping Sonification would include parameters of the mapping function. Given a data set, optimizing a sonification means to find suited parameter vectors θ that yield suitable sonifications s.

For the introduction of the technique we here use a generalized linear Parameter Mapping Sonification: for each data vector $\vec{x}^{\alpha}$ (row α in the data matrix X) an acoustic event $\phi(t; \vec{p})$ is computed using a synthesizer algorithm, so that the resulting sonification is the superposition of all individual events

$$s(X, \theta, A) = \sum_{\alpha=1}^{N} \phi(t, \vec{p}(\vec{x}^{\alpha}; \theta)) \; .$$

The d_p-dimensional synthesis parameter vector $\vec{p}$ for the synth ϕ is here computed via a linear mapping function for every d-dimensional data vector $\vec{x}$ by

$$\vec{p} = A \cdot \vec{x} + \vec{b}$$

where A is a $d_p \times d$ matrix and $\vec{b}$ a d_p-dimensional offset vector. For the selected sonification technique, the parameter vector θ would thus be the tuple $(A, \vec{b})$, and so it would have the dimension $d_\theta = d_p \cdot d + d$. The parameter vector θ takes the interpretation of a genetic code to characterize the sonification technique. We will now address the question how we can automatically create good guesses for locations in parameter space θ that yield informative sonifications.

2.3. The Quality Function for Optimization

For optimization in general it is valuable to know a quality function $Q(\theta)$ at all possible parameter vectors θ and we would aim at finding maxima of Q. If we would know Q, optimization would still be difficult since Q could exhibit multiple local optima. However, we do not have Q and suggest here a technique to iteratively construct estimates for Q from the users' relevance feedback. Since Q is used to create guesses for candidates to be heard and reviewed in the following iteration by the listener, we include besides a quality term Q_R additional function terms: (a) a novelty term Q_N is used to make unexplored regions of parameter space appealing, (b) a structural quality term Q_S is introduced in the spirit of projection pursuit techniques [6] to reward parameter regions that yield more "interesting" sonification for the given data set. Altogether, optimization will be performed in a closed loop at hand of a quality function

$$Q(\theta) = \gamma_R Q_R(\theta) + \gamma_N Q_N(\theta) + \gamma_S Q_S(\theta)$$

where the functions change on each iteration at hand of the collected feedback. The coefficients $(\gamma_R, \gamma_N, \gamma_S)$ allow to specify the subjective importance of the terms.

2.3.1. Relevance-based Quality Term

In the course of the iterative optimization, we allow the system user to rate the quality of any heard sonification on a continuous scale from -1 to 1, higher values to be given for subjectively better sonifications. We thus incrementally collect a data set $R = \{(\theta_\alpha, r_\alpha) : \alpha = 1 \ldots N_r\}$ of scalar ratings r_α in parameter space θ. We now construct a continuous relevance map in θ-space by using kernel regression with a gaussian multivariate kernel K_σ of bandwidth σ

$$Q_R(\theta, R) = \frac{\sum_{\alpha=1}^{N_r} K_\sigma(\theta_\alpha - \theta) \cdot r_\alpha}{\sum_{\alpha=1}^{N_r} K_\sigma(\theta_\alpha - \theta)} \qquad (1)$$

Thinking of Q_R as a topographic map, positive rated sonifications cause smooth hills whereas negative rated sonifications carve valleys into the landscape as shown in Fig. 2. With every new rating the relevance map is adapted to reflect all gained experience.

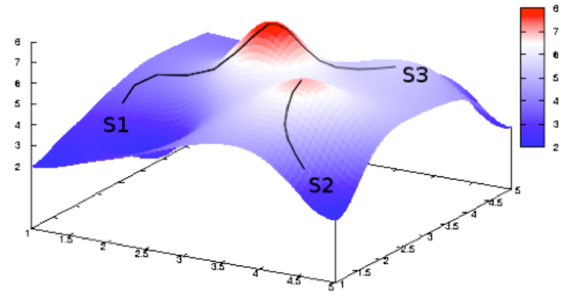


Figure 2: Illustration of a quality function as relevance map. Here the parameter vector is 2D. S1, S2, S3 show a gradient-ascend-based optimization.

2.3.2. Novelty-based Quality Term

Taking *curiousity* as principle, we would like the system to be drawn in a certain amount to those regions of parameter space where we have no idea yet how the quality is. This can be modelled by a novelty term that gives more positive answers the more distant a parameter vector is apart from already explored parameter vectors. Mathematically a novelty term can be expressed in different ways, we here suggest and use

$$Q_N(\theta; R) = 1 - 2 \cdot \exp\left(\frac{-\min_\alpha \|\theta - \theta_\alpha\|^2}{2\sigma^2}\right)$$

which increases with increasing distance from known parameter vectors asymptotically to 1.

2.3.3. Structure-based Quality Term

Quality and novelty term alone would already allow a good interactive optimization. However, the bottleneck of the approach is the human resource of providing feedback. How can we reduce the number of examples to be reviewed (and thus to accelerate the optimization) by knowledge of the optimizaton goal? Projection pursuit [6, 7] is a visual technique that automatically generates informative projections of data by maximizing a measure of *interestingness*. The objective is that high-dimensional projections tend to be gaussian distributed, and thus, the less gaussian a projection is, the more informative is it. We here introduce a quality term that measures the interestingness of a sonification at hand of a similar index. We can almost directly use structure evaluation indices from projection pursuit in our specific case since the sonification technique (linear parameter mapping) is structurally just a projection, the acoustic attributes being the projection axes. To keep the approach simple, we focus only on the temporal organization of the acoustic events. Given a parameter vector θ we obtain for the data set X a set of onset values $t_1, t_2, \ldots, t_N$. To express their structuring we use an entropy measure on the distribution of onsets. Since the onset range is always normalized to $[0, t_{max}]$, we compute a histogram of the onsets with $\sqrt{N_r}$ bins and compute the entropy by $H(\theta) = \sum_i p_i \ln(p_i)$ over all bins i where p_i denotes the relative frequency of occurrences in bin i. For our structure measure, we would like low values both for the uniform distribution (which has the highest possible entropy $H = H_{max}$) and also the most concentrated distribution (entropy $H = 0$). We thus use as structure term the quality function

$$Q_S = \frac{8}{H_{max}}\left(H(\theta) - \frac{H(\theta)^2}{H_{max}}\right) - 1$$

which is a paraboloid with a maximum at $0.5H_{max}$. The following sections will explain how the structure term is used to improve the rendering of better candidate sonifications for the next iteration.

2.4. Evolutionary Optimization

Evolutionary algorithms inherit from biological evolution to model the efficient exploration of 'genetic codes' (by mutation and sexual reproduction) that produce species of good ability to survive, often expressed as 'fitness'. In our case, the parameter vectors can be interpreted as 'chromosomes', the sonifications are the described species, and reproduction under mutation is achieved by copying the parameter vector θ and adding some random noise vector. Finally, survival means for a chromosome that its sonification is selected by the user. But even before that selection, we can introduce evolutionary techniques to create the fittest selection of candidates to be presented to the user. Assume that we need k parameter vectors to be presented to the user for the next iteration, starting with the previously selected chromosome $\theta^{(t)}$ at iteration t.

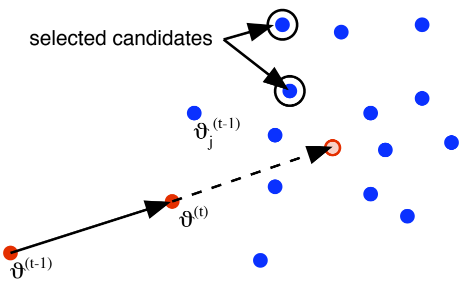


Figure 3: Chromosome reproduction with mutation in the Evolutionary Algorithm: assuming $k = 2$, 14 chromosomes are generated and the 2 with best quality $Q(\theta_j)$ are selected.

We generate $7k$ mutations θ_j as shown in Fig. 3 by

$$\theta_j^{(t+1)} = \theta^{(t)} + \sigma^{(t)} \cdot (\theta^{(t)} - \theta^{(t-1)}) + \Phi(\sigma_m^{(t)}) \,.$$

This means that we maintain a directional drift $(\theta^{(t)} - \theta^{(t-1)})$ observed from the previous two iterations, and continue it while adding some gaussian noise $\Phi(\sigma^{(t)})$ of variance ${\sigma^{(t)}}^2$ which allows us to control the influence of mutations. Stronger mutations allow a faster exploration of the parameter space, however, it may result in many chromosomes unable to survive.

Using this stochastic optimization, we select k chromosomes out of the $7k$ by taking those with maxium fitness according to our quality function Q described above. Certainly, some conditions must be checked on the generated chromosome, e.g. the parameter components are restricted to the hypercube $[-1, 1]^{d_\theta}$. Details on these additional constraints can be found in [8]. Obviously, the longer the interaction continues, the more probable it is to find suitable, interesting and novel sonifications.

The optimization depends on some hyper-parameters, namely the number k of sonifications to be reviewed every iteration, the bandwidth σ_m for mutations, and bandwidth parameters within the quality function terms. In general, it is advisable to start with rather large values and to decrease these values during the ongoing interaction according to an exponential decay schedule. Initial values can typically be set in terms of parameter and data ranges.

2.5. Gradient-based Optimization

Gradient ascend on a quality function is a frequently used approach in optimization and allows to find local optima [9]. Given a quality function Q, we can simply compute an update step $\Delta\theta$ by

$$\Delta\theta = \epsilon \cdot \nabla_\theta Q = \epsilon \cdot \left(\frac{\partial Q}{\partial \theta_1}, \ldots, \frac{\partial Q}{\partial \theta_{d_\theta}}\right)$$

At small learning rates ϵ we will move up-hill the function Q and thus create a candidate for a better rated sonification parameter vector. A key difference to the evolutionary approach is, that here the quality function is directly used to generate candidates, while in the evolutionary approach it is merely used to measure the fitness. Being potentially more direct than the stochastic search via mutation in the evolutionary algorithm above, gradient ascend faces in this form some problems: firstly, since there is only one gradient of Q, how can we generate k different suggestions for the next iteration?; secondly, gradient ascend bears the risk of getting stuck in local optima in parameter space; and finally, the quality function terms defined above are not differentiable, since they employ non-smooth functions like for instance $\min$.

The key idea for the generation of several candidates using Q is to balance the terms of Q differently for each descendant. The descendants are computed by

$$\theta_j^{(t+1)} = \theta^{(t)} + \epsilon \cdot (\gamma_{R,j}\nabla_\theta Q_R + \gamma_{N,j}\nabla_\theta Q_N + \gamma_{S,j}\nabla_\theta Q_S)$$

and thus we use diffferent vectors $\vec{\gamma}_j = (\gamma_{R,j}, \gamma_{N,j}, \gamma_{S,j})$ to define variations of quality functions that pronounce different aspects. For instance the vector $\vec{\gamma} = (0.2, 0.6, 0.2)$ mainly moves in direction of increasing novelty in parameter space. For the generation of k descendants, we use the vectors $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ for the first three candidates, followed by random mixture vectors $\vec{\gamma}_{rnd}$ normalized to length 1. As side effect we circumvent the problem of getting stuck in local optima, since all the different utilized quality functions have different optima and the functions are updated with every iteration anyway.

As mentioned before, for the computation of the gradient we need quality functions that can be differentiated. We have no problem with the Q_R as defined in eq. 1, but we need a modified novelty term to simplify differentiation. Instead of a minimum computation, we here use the average distance to all seen parameter vectors θ^α as follows:

$$Q_N(\theta) = \frac{1}{N}\sum_{\alpha=1}^{N}\left(1 - 2 \cdot \exp\left(-\frac{\|\theta - \theta^\alpha\|^2}{2\sigma^2}\right)\right)$$

For the structure term, we only replace the histogram computation by a continuous density estimation using kernel density estimation with a gaussian kernel. The mathematics gets more complex and the derivations can be found in [8], however, this allows to compute gradients in a straightforward way.

3. SYSTEM IMPLEMENTATION

The system for interactive optimization of sonification techniques has been implemented in *SuperCollider* [10] and the free mathematics package Octave[1]. For using octave code within SuperCollider, an interface class named *OctaveSC* has been developed which is available at the OctaveSC website[2] [11].

The system consists mainly of three SuperCollider classes: MasterControl, SonControl and SonOptParMap, illustrated in Fig. 4, which use a self-built Octave function library. The start script allows interactive textual specification of the data set to be used, the synthesizer code (as SC Synth) and various parameters of the optimization system. From here, a MasterControl object is instantiated with data set X, the number of descendants k per iteration, the total duration of single sonifications, the sc server to be used and a flag to select the optimization method. Optional arguments are an own synthesizer definition and a mapping function. By using such scripts, maximum flexibility is gained, and sonification types can be stored in a compact textual, human-readible form.

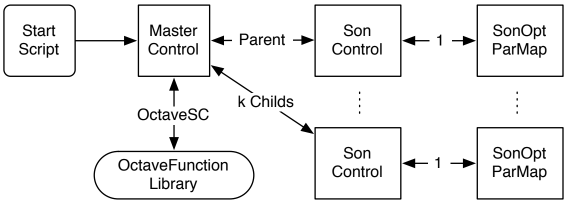


Figure 4: Diagram of the implemented software components for the Relevance-based Interactive Sonification System.

MasterControl is the main control class which manages the whole optimization process. It creates the GUI-window which encompasses some global control elements like an annealing slider for a global bandwidth parameter, optimization-specific text fields for global weights for the quality function terms in evolutionary optimization, and it furthermore hosts GUI elements for the parent/children sonifications as described next. MasterControl creates k SonControl objects which are the graphical presentations of a sonifications. Each one contains one SonOptParMap object and provides the rating slider and different buttons (play, load, save) for user interaction, depending on whether the instance represents a parent or a child. Finally, MasterControl activates the efficient vector-based computation accomplished by the Octave function library. This library contains algorithms to compute the quality function $Q(\theta)$ for both gradient-based and evolutionary optimization.

SonOptParMap encapsulates all functionality related to a single sonification, given a data set, a mapping function and a synths with d_θ parameters. The most important function is to execute the mapping function row-wise and to generate a score (as instance of a SuperCollider Score). For further details, see [8]).

For user interaction, we use the standard SuperCollider GUI elements under OS X, but using the SC GUI wrapper class by de Campo would make the porting of our system to Linux or Windows OS rather straightforward. Fig. 5 shows the main control window for interactive optimization using the evolutionary algorithm. On the left side, the user can adjust the optimization parameters $(\gamma_R, \gamma_N, \gamma_S)$, or reset the map. The centered buttons at the top allow to play the 'parent' sonification, to change its rating via the horizontal slider, or to save/load the SonOptParMap instances.

The slider on the right side depicts the actual annealing parameter: the higher the slider, the larger the jumping variance for mutation, resp. the learning rate for gradient-based optimization. The row of GUI elements below allows to trigger and review the next generation of candidate sonifications. The sliders allow a continuous quality rating and are color coded, from bad ratings (red, left) to good ratings (green, right). After reviewing the sonifications, one of them can be accepted using the 'Accept'-button. All reviewed sonifications are then integrated into the knowledge base for future quality function computations, and the accepted parameter vector is taken as 'parent' for the next iteration.

Obviously, only very little know-how is required to understand and use the interface. The simplicity of the interface favours a full concentration on the sonifications and on the listening part. Typicial sonifications suited to be explored with this technique last about some seconds so that the complete sonification fits well into short term memory and furthermore does not overly disrupt the smooth flow of interaction.

[1] see http://www.gnu.org/software/octave/

[2] see http://www.sonification.de/projects/sc

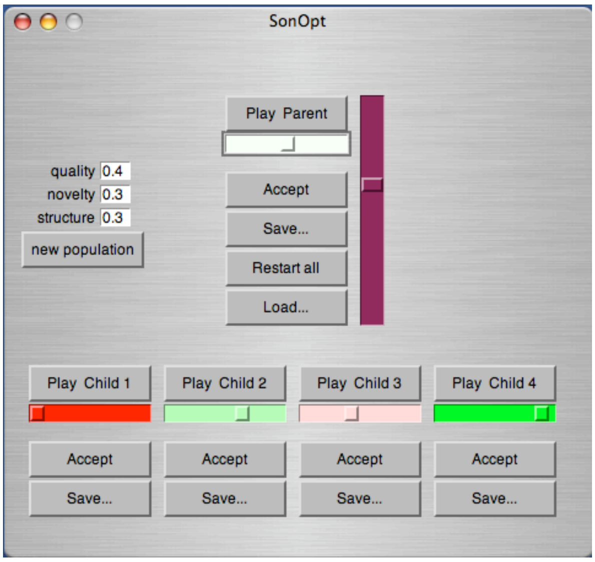

quality 0.4
novelty 0.3
structure 0.3

Figure 5: Graphical User Interface for Controlling Interactive Optimization of Sonifications, GUI for the evolutionary optimization approach.

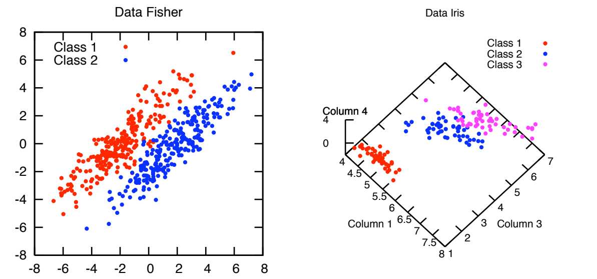
4. EXAMPLES

We will now present and discuss some examples of the interactive optimization process on the basis of different optimization methods and two different data sets shown in Fig. 6. In both examples we use discrete parameter mapping where each data item creates an audible event. Different from the general explanation above, the mapping itself is in detail a little more complex than applying an affine linear transformation. The data set is first normalized (scaled to [0,1] along features), and mapping results (which are constrained as a subset in [0,1]) are then scaled to [min, max] ranges for each attribute. Only onset receives a special treatment so that the min/max values after mapping are scaled to [0, totalduration]. The additional scalings are done for practical reasons, since some attribute values like frequency vary typically in the range of 1000 while panning ranges from -1 (left) to 1 (right).

4.1. Searching for Clustering Structure

To demonstrate basic operation of our new approach we first use a 2-dimensional benchmark data set (see Fig. 6.1), which contains of two toppled, somewhat overlapping classes, and a single-parameter sonification, where onset of granular sound events is the only parameter. Specifically we use for sound synthesis the very simple synthesizer code Out.ar(0, Pan2.ar(Blip.ar(440, 4), 0, EnvGen.kr(Env.perc(0.001, 0.1, -4), 1, doneAction: 2))), so that short harmonic sounds fill the sonification time. In result, the parameter space is 2D and visualizations of the resulting relevance map can easily be depicted without loss.

The following sonification examples accompany the stepwise assembling of a relevance map in evolutionary optimization, using the simple 1D-synths described above. Quality ratings have been given with the aim to emphasize the audible dispartment of clusters in the Fisher data set and good exploration of the parameter space. Sonification examples are available at [12].



6.1: Fisher: synthetic 2D benchmark dataset

6.2: 3D projection of Iris: 4D clustering dataset

Figure 6: 2 clustering datasets of different dimensions.

We first used the default weights of the quality function: 0.4 for quality, 0.3 for novelty and 0.3 for structure and a small step range of 0.3. The first samples were randomly generated and we heard a basically gaussian structure. This was not very informative so we assigned a negative rating and selected the parent (hear sound SF1). In this phase the relevance map was mainly affected by the novelty term and possessed wide blue regions. The few negative rated samples have drawn depth cavities in the surface (shown in Fig. 7.1). Up to step 6 we have heard no good samples and distributed exclusively bad ratings near -1 (confer figure 7.3). But in step 6 we discovered one sonification which exhibited an audible gap near the center of the sound, and we assigned a good rating and chose this as following parent (hear example SF7).

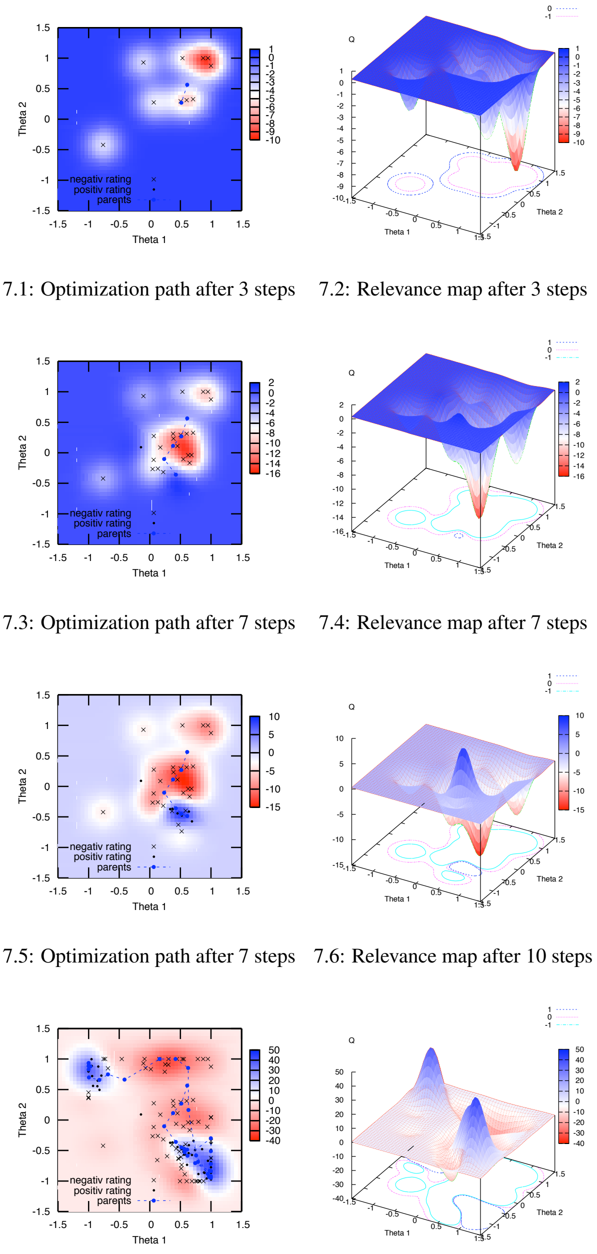
In the following steps it was possible to get a more and more audible break in the sounds: we find a maximum (hear sound SF10 and see Figs. 7.5 and 7.6). In this phase the generated sonifications exhibited not anymore significant differences. After 24 steps with no relevant modification, we now increased the range by setting the vertical slider, and in addition we set the weight of the novelty term up to 1 to better explore the entire parameter space. In result we found another maximum (Fig. 7.7) of Q, and in turn the resulting relevance map shown in Fig. 7.8 is a complete and expected representation of quality in parameter space. With sonifications of 3-4 seconds and 5 children per iteration, the inspection of one generation takes about 15-20 secs. Adding some time to specify the quality ratings, one step thus takes around 30 secs. A full optimization with 10 generations is thus completed within 5-6 mins.

4.2. Exploring Structure in the Iris data

The following example demonstrates a more realistic example. The data set is the well-known 4D Iris data set [13] (see Fig. 6.2), often used in pattern recognition and clustering. The data set consists of 3 classes, 50 instances each and 4 numeric attributes where each class refers to a type of iris plant, namely Iris Setosa, Iris Versicolor, Iris Verginica.

For the Iris data set we utilize a 7-parameter synth with onset, duration, frequency, brilliance, attack time, release time and stereo panning as acoustic attributes.

We here chose evolutionary optimization with default weights of Q set to 0.4 for quality, 0.3 for novelty and 0.3 for structure. The first set of sonifications are rendered from randomly drawn parameter vectors. They were not very interesting: a ladder of decreasing events with little attack and release time (hear SI1).



7.1: Optimization path after 3 steps 7.2: Relevance map after 3 steps

7.3: Optimization path after 7 steps 7.4: Relevance map after 7 steps

7.5: Optimization path after 7 steps 7.6: Relevance map after 10 steps

7.7: Optimization path after 37 steps 7.8: Relevance map after 37 steps

Figure 7: Visualization of an optimization pass using the Fisher data set after 3 interaction steps (7.1), 7 steps (7.3), 10 steps (7.5), 37 steps (7.7) and a 3D visualization of the relevance map after 10 steps (7.6) and 37 steps (7.8). Color indicates the quality of the parameters in space θ from good (blue) over neutral (white) to bad (red) as approximated by the quality function $Q(\theta)$.

The second sound is divided into 2 parts: a mixture of events with little attack and release time of akin frequency in middle up to left direction and a second stage of mixed events (hear SI2). The first impression leads us to a assumption of 2 separated classes. In the following samples we tried to increase the differences of the rattle. After 2 iterations we found a sonification, that contains some events with a higher brightness (number of harmonics) at the end of the sound, so that they got a ”blob”-like ringing (SI4). Then in step 7 we first hear a division of 3 classes: the first and second class are divided by a small intermission, whereas the third class is represented by both different frequency and number of harmonics (SI7). Finally this trait was increasingly optimized until step 10 to become more salient (SI10).

This is only one way to manifest structure in the iris data. We can also accomplish the division with other parameters, like stereo panning or amplitude, as demonstrated in the sonification examples SPan and SAmp which show the results of an active optimization towards the salience of structure in these attributes. Generally, the classes are represented by the mixture of different parameters, but it is possible to maximize the influence of one targeted acoustic parameter during optimization. The advantage is that each user, depending on his/her own hearing skills, can optimize the sonification as preferred to extract maximal information from the sound about the data within some minutes time.

5. DISCUSSION AND CONCLUSION

We have introduced a novel approach for interactively optimizing sonifications by reviewing iteratively derived sonifications. Conceptually relevant, our approach decouples the skills of programming sonifications and judging sonifications. Whereas past approaches often require the user to bring together knowledge about the parameters or the sonification technique, this approach frees the user from such burden, and allows a clear focus on the listening part. In result, the user acts like a navigator who merely controls the course of optimization flow by high-level feedback, guiding the system to parameter space regions with attractive perceptual qualities.

In this paper, we have presented a special implementation of this idea using the sonification technique of Parameter Mapping Sonification (PMS), which is still the most wide-spread sonification approach. While, however, in many practically seen (better: heard) PMSs, data features and acoustic parameters are connected one-to-one, our generalized approach allows arbitrary linear mixtures. We'd like to emphasize that the technique is not only capable of optimizing general PMSs, but also other types of sonifications, such as audifications, Model-based Sonifications, or even Parameterized Auditory Icons.

A key element in our optimization system is the successive storage of all considered candidates' parameter vectors together with their quality rating. This does not only allow the system to compute a relavance map to predict the quality of any randomly drawn sonification parameters, it furthermore allows – as an analysis target beyond the scope of this paper – to better understand human listening and the relation of complex auditory stimuli to their capabilitiy to evoke human perceptions of structured qualities. For instance, consider that the user is asked to actively create sonifications where data clusters emerge as perceptual clusters. By using our approach it can be found to what degree acoustic parameters best contribute to the creation of this perceptual high-level quality. While such applications of our system actually represent more a side-effect than the main goal, we regard the use of interactive sonification as an experimental technique for cognitive analysis of human listening to complex stimuli as highly attractive.

Concerning the optimization, we have demonstrated two complementary approaches for creating useful candidates: (a) an evolutionary approach which generates descendents (or their parameter vectors which might be called 'chromosomes') via mutation: random changes, followed by a pre-selection of the best (fittest) descendants give rise to a new generation of sonification to be reviewed by the user, and (b) gradient-based optimization, where the different candidates follow the gradient in different variants of the quality function created by weighting functional terms specifically. Thus, there is always a candidate with optimal novelty, quality, and structure to be reviewed in the set. Both approaches are useful, and while the latter one may be more goal-oriented in settings where the quality map is in fact rather smooth, the first approach may be superior for complex settings and a stronger exploration.

The reader may have the impression that such sonifications are so strongly tuned to the subjective preferences of the user that they may not be particularly 'objective' to communicate structural features in the data. However, sonification is actually always the result of strongly subjective tuning of parameters. Furthermore, each mapping is equally valid as true representation of the data. Only the combination of different (sonic) 'views' may yield a more 'objective' overall impression of structures in the data. The following example may show in what way the optimization is highly useful besides the subjective exploratory value for the subject: Consider for instance the optimization of *perceptual contrast* between the first and second half of a EEG time-series sonification, e.g. for a situation where some particular event occurs. The maximization of perceptual contrast may yield sonification parameters that are suited to be used with other data, e.g. to discover similar events in non-labeled data reviewing. Thus the aim is not to 'mis-tune' one sonification in order to express whatever you like, but to create highly specific, contrast-maximizing sonifications that are likely to operate particularly well for other data sets of comparable typology.

Finally, we have demonstrated the system's capability to find structure in a low-dimensional clustering benchmark data set, so that the relevance map can easily be visually inspected, and shown operation on a real-world data set. We currently work towards real-world application of *Interactive Contrast Maximization* for complex time series, and we particularly continue our research towards the development of more tightly coupled human-computer interactive sonification systems by means of continuous interactions.

Acknowledgement

We thank Christian Mertes and Thorsten Stocksmeier at Bielefeld University for their programming work in context of the project seminar 'Optimization of Sonification Techniques'.

6. REFERENCES

[1] C. Scaletti, "Sound synthesis algorithms for auditory data representations," in *Auditory Display*, G. Kramer, Ed., Reading, MA, 1994, pp. 223–248, Addison-Wesley.

[2] Thomas Hermann, Jan M. Drees, and Helge Ritter, "Broadcasting auditory weather reports – a pilot project," in *Proceedings of the International Conference on Auditory Display (ICAD 2003)*, Eoin Brazil and Barbara Shinn-Cunningham, Eds., Boston, MA, USA, 2003, International Community for Auditory Display (ICAD), pp. 208–211, Boston University Publications Production Department.

[3] G. Kramer, "An introduction to auditory display," in *Auditory Display*, G. Kramer, Ed. ICAD, 1994, pp. 1–79, Addison-Wesley.

[4] Thomas Hermann, *Sonification for Exploratory Data Analysis*, Ph.D. thesis, Bielefeld University, Bielefeld, Germany, 2 2002.

[5] P. Runkle, A. Yendiki and G. H. Wakefield, "Active Sensory Tuning for Immersive Spatialized Audio," in *Proc. Int. Conf. on Auditory Display (ICAD 2000)* P.R. Cook (Ed.), Atlanta, GA, 2000, pp. 141-145.

[6] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. 23, pp. 881–890, 1974.

[7] P. J. Huber, "Projection pursuit (with discussion)," *Annals of Statistics*, vol. 13, pp. 435–525, 1985.

[8] Kerstin Bunte, "Interaktive relavanzkartenbasierte Optimierung der Sonifikation von Daten," Diplomarbeit, Bielefeld University, Bielefeld, Germany, Sepp., 2006.

[9] B. D. Ripley, *Pattern Pecognition and Neural Networks*, Cambridge University Press, 1996.

[10] James McCartney, "Supercollider: a new real time synthesis language," in *Proc. ICMC '96*. Int. Comp. Music Assoc., 1996, http://www.audiosynth.com/icmc96paper.html.

[11] Thomas Hermann, "OctaveSC - a SuperCollider interface class for Octave," http://www.sonification.de/projects/sc3/index.shtml, Mar. 2006.

[12] Thomas Hermann, "sonification.de - Publications - enline examples" `http://www.sonification.de/publications`, 2007.

[13] R. A. Fisher, "UCI repository of maschine learning databases," `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris`, 1999.