

KERNEL REGRESSION MAPPING FOR VOCAL EEG SONIFICATION

Thomas Hermann¹, Gerold Baier², Ulrich Stephani³, Helge Ritter¹

¹ Neuroinformatics Group, Bielefeld University, Germany
thermann@techfak.uni-bielefeld.de

² Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, UK; and
Facultad de Ciencias, Universidad Autónoma del Estado de Morelos, Cuernavaca, Mexico

³ Neuropediatric Clinic, University of Kiel, 24105 Kiel, Germany

ABSTRACT

This paper introduces kernel regression mapping sonification (KRMS) for optimized mappings between data features and the parameter space of Parameter Mapping Sonification. Kernel regression allows to map data spaces to high-dimensional parameter spaces such that specific locations in data space with pre-determined extent are represented by selected acoustic parameter vectors. Thereby, specifically chosen correlated settings of parameters may be selected to create perceptual fingerprints, such as a particular timbre or vowel. With KRMS, the perceptual fingerprints become clearly audible and separable. Furthermore, kernel regression defines meaningful interpolations for any point in between. We present and discuss the basic approach exemplified by our previously introduced vocal EEG sonification, report new sonifications and generalize the approach towards automatic parameter mapping generators using unsupervised learning approaches.

1. INTRODUCTION

Multivariate time series is a frequent data type in many scientific contexts, and particularly in biomedical applications, such as EEG, EMG, ECG, fMRT, to name a few. The high-dimensionality poses a particular challenge to understand the structure of the state space, and furthermore the dynamical aspects which manifest themselves in the time domain in the form of rhythm, rhythmic changes, phases between channels and their systematic change. The traditional way (still regularly used in clinical practice) of exploring such features of alike data is by plotting the parallel time series as shown in Fig. 1. This allows direct comparison of features and their changes in the individual time series. However, it remains difficult to interpret systematic dependencies between different channels from the plots, e.g. do the phase relations between the channels change over time or do they stay constant? We are particularly interested in a meaningful auditory representation of the human electroencephalogram (EEG). As a novel approach to achieve this, we intro-

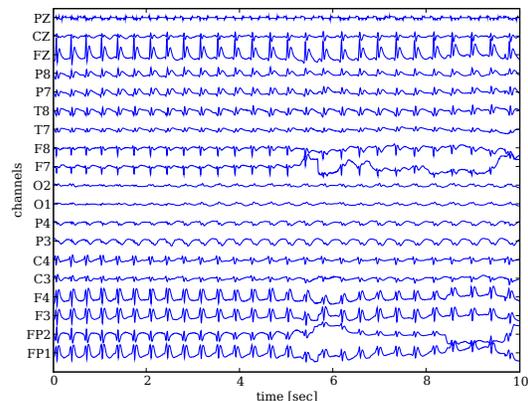


Figure 1: Typical plot for reviewing high-dimensional time-series data. Shown are 19 channels of a standard EEG during an absence seizure.

duced the technique of Vocal EEG Sonification [1] to render sonifications so that characteristic spatio-temporal patterns (or motif sequences) in the data lead to corresponding patterns of vowel-transitions. From these sonifications we obtained structured auditory gestalts as emergent features. The gestalts might be coined *acoustic fingerprints* or *signatures*. Acoustic fingerprints would be of clinical interest if they could be shown to correspond to known and discernible pathologies in the data. Such a spontaneous emergence of auditory gestalts is a significant advantage of our technique over other types of parameter-mapping sonifications and it connects well with the capability of the human auditory system to constitute perceptible gestalts and recognize them if they occur repeatedly.

The selection of vowel-like sounds and – as a consequence of time-dependent data – vowel transitions, was motivated by the fact that human listeners are already highly adapted to the segmentation and interpretation of similarly structured patterns from processing speech signals. Fur-

thermore, since we all are able to generate speech sounds with our own articulatory system, we can communicate patterns by imitating them vocally – which is an ideal condition for collaborative examination of EEG data, because one researcher can directly draw a colleague’s attention to a pattern by imitating it.

In our original implementation of Vocal EEG Sonification [1], we paid special attention to achieve generality in the sense that sonifications could be rendered even with different recording conditions like a different number or position of channels, or a different sampling rate. We computed the dipole $q = (q_x, q_y)$ of electric activity on the scalp and mapped the x/y -components of the dipole vector, corresponding to hemispheric and anterior-posterior disbalance, to the first two formant frequencies of a subtractive synthesizer. Since vowels are mainly characterized by their first two formants as shown in Fig. 2, this produced transitions between vowel sounds.

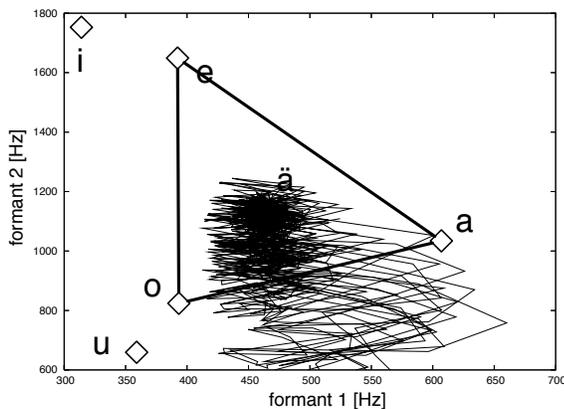


Figure 2: Plot showing vowel sounds in the space of the first two formants. The plot is reproduced from our previous paper [1].

However, the perceptual quality of the synthetic vowels was limited. In natural speech signals the coordinated movement of several formants, including their center frequencies, their bandwidths and their gain play together to shape the timbre, and thus the parameters governing those characteristics show complex dependencies. We experimented with mappings from various data features to more than two formants but with limited or no improvement of the saliency. This motivated us to rethink the mapping mechanism in search for a mapping that could create clearly distinguishable sounds of controllable articulation accuracy.

In this paper we present a kernel-regression based approach to mediate between high-dimensional data spaces and high-dimensional parameter spaces for sonification so that specific acoustic constellations are reached at con-

trollable conditions, and additionally a controllable interpolation is achieved in data space. We describe the technique and exemplify the approach with the vowel-creating sonification described above. As a generalization of this approach, we then present a scheme to combine kernel-regression based mapping with unsupervised learning techniques such as vector quantization to achieve semi-automatic data-driven mediators for sonification.

2. KERNEL REGRESSION MAPPING

In Parameter-Mapping Sonification, the most frequent form is the one-to-one mapping of data variables to acoustic parameters, such as for example mapping channel i th data $x_i(t)$ to pitch $p(t)$. Different mapping functions such as linear, exponential, sigmoid, etc. have been used for this basic case.

A more general way to compute acoustic parameters \vec{p} is to allow mixtures of several data features \vec{x} , e.g. $p_1 = \sigma(\sum_i a_i x_i)$ with mixing coefficients a_i and a mapping function $\sigma(\cdot)$, or in matrix form:

$$\vec{p} = \vec{\sigma}(\mathbf{A}\vec{x} + \vec{b}) \quad (1)$$

This linear mixing followed by optional nonlinear mapping, however, does not support an association of different parameter vectors \vec{p}^α to specific locations in data space \vec{x}^α , our main goal as described in Sec. 1.

Kernel regression is a standard approach to compute smooth interpolations between given output vectors [2], and we here show how it can be used to create a new family of kernel-based sonification mapping techniques. Without loss of generality, we focus on one-dimensional outputs – high-dimensional outputs are then obtained by using parallel kernel regression units for each vector component.

Kernel regression computes an output value p for an input x by averaging the outputs of given prototypes p^α , located at inputs \vec{x}^α according to the strength of their responsibility to contribute to the value at position \vec{x} , which is quantified by a kernel function $K(\vec{x}, \vec{x}^\alpha)$. Using linear kernel functions deliver the above mappings as a special case. A typical choice for interpolation are Gaussian kernels

$$K_\sigma(\vec{x}, \vec{x}^\alpha) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\vec{x} - \vec{x}^\alpha\|^2}{2\sigma^2}\right) \quad (2)$$

where the bandwidth parameter σ controls the region of influence of a given prototype.

The interpolation result is obtained by

$$p(\vec{x}) = \frac{\sum_\alpha K_\sigma(\vec{x}, \vec{x}^\alpha) \cdot p^\alpha}{\sum_\alpha K_\sigma(\vec{x}, \vec{x}^\alpha)} \quad (3)$$

At very large values of σ the output becomes approximately the mean of all p^α for all inputs. The smaller σ , the better separated are the outputs. At position \bar{x}^α , the output p^α dominates increasingly with decreasing σ . Thereby we gain a smooth interpolation of outputs for inputs between the prototypes.

A one-dimensional example is depicted in Fig. 3, to help to understand the basic operation of kernel regression. The calculated parameter value is a smooth function for large values of σ and approaches a discrete set of points as $\sigma \rightarrow 0$. For high-dimensional outputs, as in case of mappings

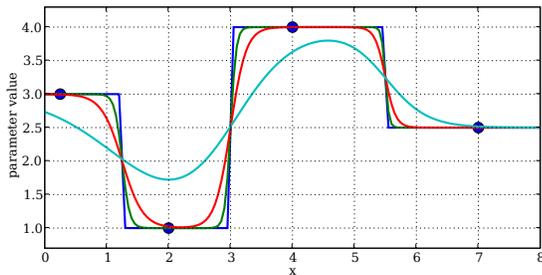


Figure 3: Illustration of kernel regression using a Gaussian kernel of bandwidth 0.1 (blue), 0.3 (green), 0.5 (red) and 1 (cyan). Prototypes are marked as blue circles. Note that the maximal fit of output values does not necessarily coincide with the prototype x -value.

on parameter vectors, each output vector component is the result of a kernel regression. Thus a trajectory in data space passing from one prototype position \bar{x}^α to another prototype \bar{x}^β will lead to a trajectory in parameter space that moves slowly near \bar{p}^α and near \bar{p}^β but much faster in between, if σ is much smaller than the prototype distance $\|\bar{x}^\alpha - \bar{x}^\beta\|$ (see Fig. 4). In the limiting case $\sigma \rightarrow 0$ we obtain a segmentation of the input space into so-called Voronoi cells of constant output vectors corresponding to the output of the nearest prototype (winner-takes all).

With this background we can now formulate Kernel Regression Mapping Sonification (KRMS) as a general process as shown in Fig. 4: in the first step some adequate *features* are computed from the data vectors. These may range from a simple selection of variables to non-local aggregate functions such as an estimation of activity in a certain frequency band. Kernel regression is then used as described above to render appropriate parameter vectors which are subsequently fed into the sound synthesis engine. This sonification scheme can be used for different types of mappings such as discrete or continuous parameter mapping sonifications and even event-based sonification.

The bandwidth parameter of the kernel regression is an intuitive control parameter to adjust the conciseness of

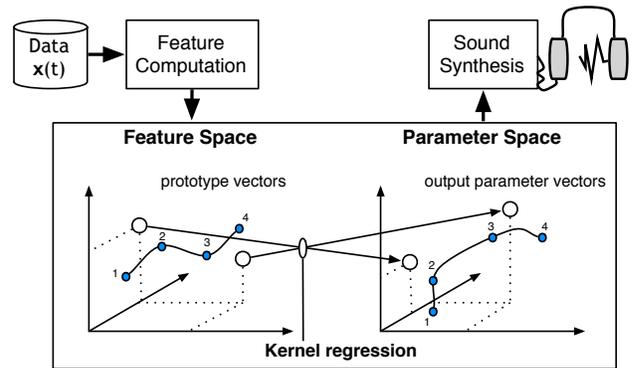


Figure 4: Illustration of the full kernel regression mapping sonification chain, which mediates between potentially high-dimensional features and parameter spaces. For multivariate time series a feature state trajectory is transformed into a continuous parameter trajectory as illustrated here for two kernel centers.

the mapping – from very smooth interpolation mappings at large σ levels to sharp transitions between prototype parameter vectors at low values of σ . Since the whole sonification chain can be processed in real-time, this parameter can also be adjusted interactively.

In the following section we exemplify KRMS in a specific biomedical application, namely the sonification of EEG data.

3. A VOCAL EEG SONIFICATION WITH KRMS

As explained in the Introduction, Vocal EEG Sonification aims at the emergence of temporally structured dynamic gestalts (or fingerprints) that characterize pathologic dynamics in the measured brain activity. The data are d -dimensional vectors (with d commonly between 19 and 40) obtained from electric potential measurements at distinct locations on the scalp, measured against some reference at a rate of 200 Hz or higher. The measurements can be interpreted as a trajectory moving in the d -dimensional state space. In our original approach, we used the hemispheric and anterior-posterior disbalance as generic data features for the mapping to vocal sounds.

In the paper we focus on a mapping of data features to acoustic parameters that are responsible for the perception of vowels: formant frequencies; formant bandwidths; and gains of a sum of filter outputs on an excitatory source signal. Details of the improved mapping data features to other parameters for the recognition of pathologic EEG features will be reported elsewhere.

3.1. The original data features

Fig. 5 shows the SuperCollider code for our vowel synthesizer. A transition between unvoiced and voiced speech is achieved by controlling vn , an additional vibrato in the fundamental f_0 is controlled by rate v_{freq} and intensity v_{mod} . Other arguments are self-explanatory, f_i , bw_i , g_i refer to formant frequency, bandwidth and gain of the i -th formant.

```
SynthDef("CV2", { | out=0, f0=135, level=0, vn=1,
  vfreq=0, vmod=0.2, pan=0,
  f1=650, bw1=80, g1=(0),
  f2=1080, bw2=90, g2=(-6),
  f3=2650, bw3=120, g3=(-7),
  f4=2900, bw4=130, g4=(-8),
  f5=3250, bw5=140, g5=(-22),
  filtcf=1000, lg=0.05 |
var ffreq, sum, av0, an0, ain;
ffreq = SinOsc.ar(vfreq, 0, mul: vmod, add: f0);
an0 = LPF.ar( WhiteNoise.ar(4), 18000);
an0 = BPF.ar( an0, f1, (bw1/f1+0.1), g1.dbamp)
+ BPF.ar( an0, f2, (bw2/f2+0.1), g2.dbamp)
+ BPF.ar( an0, f3, (bw3/f3+0.1), g3.dbamp)
+ BPF.ar( an0, f4, (bw4/f4+0.1), g4.dbamp)
+ BPF.ar( an0, f5, (bw5/f5+0.1), g5.dbamp);
av0 = Formant.ar( ffreq, f1, bw1, 100, 0.5)
+ Formant.ar( ffreq, f2, bw2, g2.dbamp)
+ Formant.ar( ffreq, f3, bw3, g3.dbamp)
+ Formant.ar( ffreq, f4, bw4, g4.dbamp)
+ Formant.ar( ffreq, f5, bw5, g5.dbamp);
ain = (vn.lag(lg)*av0)+(1-vn.lag(lg))*an0;
sum = LPF.ar(ain, filtcf);
Out.ar(out, Pan2.ar(sum, pan, level.dbamp))
}).load(s);
```

Figure 5: SuperCollider code for the vocal synthesizer used in the sonifications. For better readability the ".lag(lg)" suffix is removed in all BPF.ar and Formant.ar control arguments.

The perceived timbre (vowel sound) is at fixed fundamental frequency and voice-noise ratio mainly dependent upon the 3×5 parameters for the 5 formants. In fact, the first two formant frequencies f_1, f_2 suffice to create perceptible vowels. If we map data to these two formants, the question arises which data channels to choose. If we map data channels to all 15 parameters we find a huge number of possibilities for the mapping with mostly poor performance in the sense that they do not create clearly audible shapes. Thus, the question of mapping is a non-trivial task. For that reason we suggested in [1] to first compute meaningful features from the raw data that can then be mapped to formant parameters. As mentioned before we used the x/y-components of the electric dipole since they can be defined for arbitrary electrode montages.

Sound Example S1¹ gives an impression of vowel tran-

¹sound examples are provided at <http://sonification.de/publications/HermannBaierStephaniRitter2008-KRM>

sitions obtained. There are recognizable but vaguely differentiated vowels as in 'pot' or 'bar', and the epileptic activity leads to a periodic rhythm similar to 'how-yaaa how-yaaa how-yaaa'. Quite often the 'shwa' sound² is produced. This is plausible due to the normalization of channel data to mean 0 and variance 1 which leads to an average dipole of 0. Fig. 2 shows a typical trajectory in the 2d-parameter space of the first two formants where the center of mass lies in the middle of the vowel triangle e-o-a. Obviously in the shown data the richer part of the formant space with 'i' as in bee and 'u' as in zoo is not covered, so these vowels do not contribute to differentiate dynamic transitions in the data.

We experimented with mapping higher moments (multipoles) to higher formants without achieving an improvement of the acoustic quality or increased insight from the sonification. Now we aim at better principled, and clearer audible usage of the formant parameters, thereby diminishing the arbitrariness of such mappings.

3.2. Delay Embedding Features

To solve the mapping problem, we searched for a data feature that avoids the contraction to the 'shwa' location and suggest a two-dimensional *delay embedding* of the average potential, as explained below.

Absence seizures and related generalized epileptic activity display EEG patterns where correlated activity is picked up at distributed locations of the scalp. This leads to the generally observed global increase in correlation during such events [3]. The sum of all channels $s(t)$ therefore averages out random activity to some extent and pronounces the collective part of the activity. As such it is a well-suited feature for the purpose of mapping to formants. In case of typical absence dynamics, with a main frequency at about 3 cps, the corresponding structure is well captured in a scatter plot of $s(t)$ against its time-delayed version $s(t - \tau)$. A good choice for the delay τ is a 1/4-fraction of the shortest wave that is to be resolved – in case of a spike-wave complex at 3 Hz, where the shorter wave, so-called spike, lasts approximately 1/3 of the complete period we thus get

$$\tau = \frac{1}{3 \text{ Hz}} \cdot \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{36} \text{secs} \quad (4)$$

as a suitable choice. Fig. 6 (right side) shows the resulting scatter plot for epileptic activity. The rhythmic pattern leads to recurrent structures along narrow paths in the plot which is more obvious from the delay embedding than from the dipole components in Fig. 6 (left side). The delay-embedding proved informative for a variety of epileptic data sets and we adopted it as 2d-feature for subsequent mapping on vowels.

²neutral middle vowel, occurs in unstressed syllables

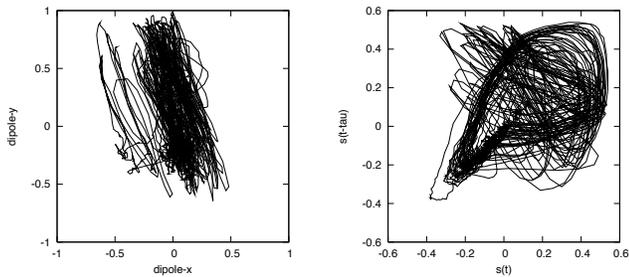


Figure 6: Data features for Formant Mapping: the left plot shows the previously used dipole vector, the right plot the new delay embedding feature - a clear recurrence structure can be recognized.

Sound example **S2** (see [4]) is a sonification where the embedding feature is directly mapped to the first two formants of the vowel synthesis as described in [1].

3.3. KRMS for Vocal EEG Sonification

Although the above motivated two-dimensional delay embedding feature offers a better occupation of the feature space than the dipole, it is still not ideal to directly map it on the first two formants. On the one hand, still most of the activity leads to transitions in the vowel triangle a-e-o so that the bright individual vowels 'i' and 'u' are rarely touched. On the other hand, as before it neglects the other $15 - 2 = 13$ parameters that are apparently very useful for the perception of concise and clear vowels.

Therefore, we now use KRM to mediate between the low-dimensional (2d) data feature space and the 15-dimensional formant parameter space. In such a situation KRMS can play out its strength, since it allows a predefined placement of prototypes in the feature space, and it furthermore delivers a continuous mapping into the 15-dimensional formant parameter space so that all vowels are produced in their cleanest form.

We layed out five vowel prototypes (a-e-i-o-u)³ in form of their corresponding 15 parameters on a pentagon into the delay-embedding feature-space as shown in Fig. 7. A trajectory that passes nearby these prototypes (and therefore induce the corresponding parameter sets) will thereby lead to perceptible transitions between the pure vowels.

The following sound examples **S3.1–S3.5** (examples at [4]) illustrate KRMS in this application by means of a series of sonifications with different bandwidth parameters, all using the same EEG dataset. In the series from **S3.1** to **S3.5** it can be heard that the transitions between formants become successively sharper with decreasing bandwidth σ .

³like in car-edge-ear-for-zoo

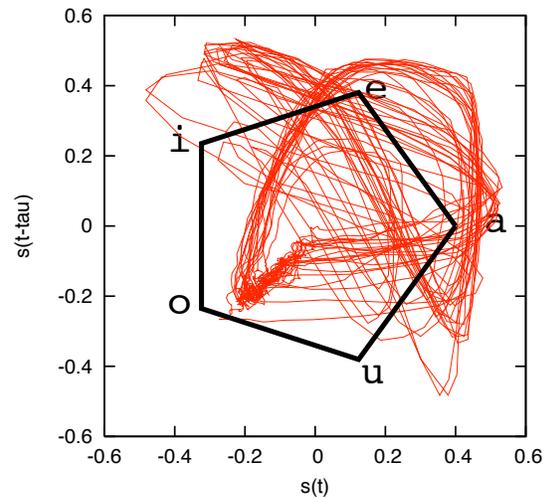


Figure 7: Vowel embedding for Vocal EEG sonification KRMS mapping. The trajectory shows the data for the rendered sonication examples S3.1-S3.5.

The sonifications were rendered at a compression of 0.5, i.e. half of real-time rate. This rate is ideal to differentiate vocal rhythms. However, for the sake of getting used to the sound we also provide examples at a compression rate of 0.25 which allow more time to attend to the vowel changes. The sounds **S4/S5.1–S5.5** correspond to sounds **S2/S3.1–S3.5**.

Figure 8 shows the resulting movements in formant space visually for different values of σ . The five center frequencies of the formants are shown as a function of time (bandwidths and gains are interpolated accordingly). For clarity, only a few oscillations are shown for each bandwidth value.

As a result, the KRM is an efficient means to obtain better perceptible and more concise vowels for a given EEG feature than a direct mapping on formants. In the application shown, the prototypes have been manually placed to obtain the sonification. In the following we suggest unsupervised learning techniques to automatically render concise sonifications for arbitrary high-dimensional data sets.

4. FINDING SUITABLE PROTOTYPE LOCATIONS

We have used KRM as a technique to anchor specific mappings locally to input space (resp. feature space) with the additional ability to automatically create useful interpolations between given prototypes. We have shown in the previous section how this feature can be used to create salient transitions between vowels.

But how can KRM be useful in the general case of an high-dimensional data space without any clear motiva-

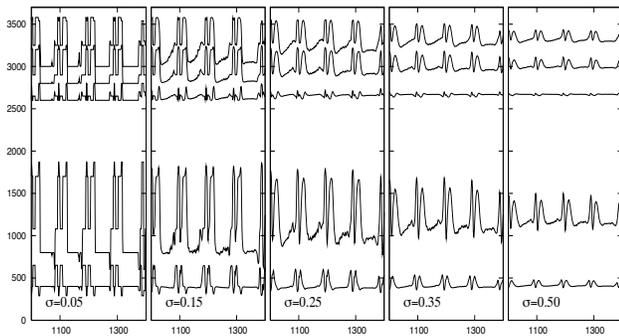


Figure 8: Formant frequencies resulting from KRMS of some few epileptiform signal periods (shown also in Fig. 6). The time axis show sample number recorded at 256 Hz, y axis shows frequency in Hz. From left to right: $\sigma = 0.05, 0.15, 0.25, 0.35, 0.50$

tion where and how to place the prototypes? In this case it makes sense to consider data-driven techniques to automatically render a limited set of prototypes that characterize the distribution in input space. The branch of unsupervised learning (as part of machine learning and neural networks) offers manifold techniques to create such representations [5, 6]. The most straightforward methods are vector quantization (VQ) where a set of prototypes is adapted to minimize the quantization error when representing the data by their nearest vector, or the self-organizing map (SOM) [7] which does approximately the same and additionally delivers a topologically ordered set of prototypes (e.g. on a grid).

Fig. 9 shows a scatter plot for the EEG features together with some prototypes that resulted from VQ-learning prototypes with all available state vectors. Obviously, a trajectory will now pass through the Voronoi cells of several prototypes and thereby create a specific sequence of sounds during a cycle. If the overall pattern in the data changes, the sequence of sounds changes automatically and allows thereby the recognition of dynamic characteristics.

By using the bandwidth parameter σ , the user has the control to navigate continuously between a more symbolic sonification where the time series is automatically decomposed in the corresponding sequence of prototype sound patterns (e.g. distinct vowels, or different pitches, if a certain pitch would be associated with a prototype) to an analogous representation. This offers a convenient way to explore the analogic-symbolic continuum discussed by Kramer [8], where changes in the time series correlate to corresponding changes in the audible form. In result, patterns in the time series translate to corresponding sequenced patterns in the sonification, with the potential to facilitate the learning and remembering of patterns, or to discover new patterns.

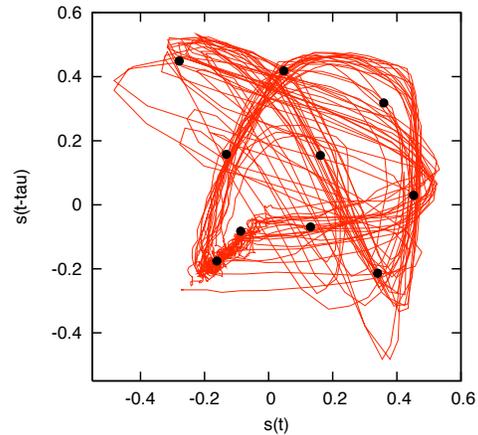


Figure 9: Vector-Quantization on delay-embedding features to adapt 10 prototypes automatically. The trajectory is segmented in a sequence of distinct auditory symbols that form a repeating rhythm. VQ prototypes are shown as circles.

5. DISCUSSION AND CONCLUSION

We introduced Kernel Regression Mapping Sonification (KRMS) as a new method to connect data spaces to possibly complex parameter spaces for Parameter Mapping Sonification. The connection is achieved via a feature representation and kernel regression-based interpolation scheme that is new in the context of sonification. Different from existing mapping schemes, KRMS represents a local method in the sense that localized points in input space are connected with a specific output. In result, mappings can be obtained that are not possible with the typical linear mixing mappings that dominate in the field.

KRMS exploits the bandwidth σ as an intuitive control parameter that allows to select the granularity of acoustic presentation on a scale from a segmentation into a discrete set of sounds to a continuous interpolation between prototype mappings. Thereby the user can set a focus on either highly accurate rhythmic details (on the symbolic side) or on a continuously variable sound shape (on the analogical side) or anywhere in between in Kramers continuum.

KRMS was motivated by our interest in canonic mappings between EEG data and vowel transitions so that the multiple parameters of vowels (5 formants, 5 bandwidth and 5 gains) are altogether coherently adjusted with the time-varying data. The sonification examples demonstrate the obvious (or: audible) superiority to render concise vowel transitions. We have not commented on the other audible changes in the Vocal EEG sonification examples such as panning, pitch changes, voice/noise ratio, and the audible gaps that have been intentionally and fully data-driven introduced to lead to the perception of consonants. These will

be addressed in a separate contribution.

KRMS is real-time capable and flexible in all relevant dimensions: mapping outputs, prototypes, and bandwidth can all be manipulated on-line if necessary, the computational effort is modest.

Other applications of sonification will likely benefit from the KRMS approach. For instance, the results of clustering algorithms can be reviewed by setting cluster centers as prototypes for KRMS and choosing a set of output parameter vectors of very different timbre. In result, KRMS will lead to sonifications where the spreading of data around the cluster centers can be judged from the deviations of the prototype sound. In this case the feature computation is not utilized, and both data and parameter space are high-dimensional. Alternatively, think of a warning system for process monitoring (e.g. of sensor measurements for a chip production line). We can easily set prototypes for positive and negative states into feature space and thereby quickly obtain a sonification that smoothly and automatically interpolates towards the sound parameters associated to the 'negative' conditions whenever such states occur. This is a behaviour that would not be easy to obtain in direct parameter mapping.

Abstracting from the specific application of Vocal EEG sonification focused here, we see KRMS as a novel paradigm for segmenting multivariate time series into sequences. Together with the available powerful techniques from machine learning / unsupervised learning we expect a variety of innovations in domains like process monitoring (e.g. [9]) and biofeedback systems (e.g. [10]).

In conclusion, Kernel Regression Mapping Sonification (KRMS) opens new avenues how to mediate between high-dimensional data spaces and often equally high-dimensional parameter spaces for parameter mapping sonification, particularly in those situations where the auditory structure created by the parameters makes a correlated and coherent control of parameters necessary, such as demonstrated for the example of vowel transitions. Our ongoing research is now directed at both the search for better ways to support the understanding of specific biomedical signals such as EEG and at the identification of basic grounding principles for successful sonification.

6. REFERENCES

- [1] Thomas Hermann, Gerold Baier, Ulrich Stephani, and Helge Ritter, "Vocal sonification of pathologic EEG features," in *Proceedings of the International Conference on Auditory Display (ICAD 2006)*, Tony Stockman, Ed., London, UK, 06 2006, International Community for Auditory Display (ICAD), pp. 158–163, Department of Computer Science, Queen Mary, University of London.
- [2] M. P. Wand and M. C. Jones, *Kernel Smoothing, Monographs on Statistics and Applied Probability*, Chapman & Hall, New York, 1995.
- [3] Gerold Baier, Marcus Müller, U. Stephani, and H. Muhle, "Characterizing correlation changes of complex pattern transitions: the case of epileptic activity," *Phys. Lett. A*, vol. 363, pp. 290, 2007.
- [4] Thomas Hermann, "Online sonification examples," <http://sonification.de/publications>.
- [5] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
- [6] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
- [7] Teuvo Kohonen, *Self-Organizing maps*, Springer Series in Information Sciences Vol. 30. Springer, Berlin, Heidelberg, New York, third edition, 2001.
- [8] G. Kramer, "An introduction to auditory display," in *Auditory Display*, G. Kramer, Ed. ICAD, 1994, pp. 1–79, Addison-Wesley.
- [9] A. Roginska, E. Childs, and M. K. Johnson, "Monitoring real-time data: a sonification approach," in *Proceedings of 12th International Conference on Auditory Display ICAD2006*, London, 2006, ICAD, p. 176.
- [10] Thilo Hinterberger and Gerold Baier, "POSER: Parametric orchestral sonification of eeg in real-time for the self-regulation of brain states," *IEEE Trans. Multimedia*, vol. 12, pp. 70, 2005.