Interactive Computation of Timbre Spaces for Sound Synthesis Control

Stefano Fasciani

Digital Signal Processing Lab, School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore

stefanofasciani@stefanofasciani.com

Abstract

Expressive sonic interaction with sound synthesizers requires the control of a continuous and high dimensional space. Further, the relationship between synthesis variables and timbre of the generated sound is typically complex or unknown to users. In previous works, we presented an unsupervised mapping method based on machine listening and machine learning techniques, which addresses these challenges by providing a low-dimensional and perceptually related timbre control space. The mapping maximizes the breadth of the explorable sonic space covered by the sound synthesizer, and minimizes possible timbre losses due to the low-dimensional control. The mapping is generated automatically by a system requiring little input from users. In this paper we present an improved method and an optimized implementation that drastically reduce the time for timbre analysis and mapping computation. Here we introduce the use of the extreme learning machines for the regression from control to timbre spaces, and an interactive approach for the analysis of the synthesizer sonic response, performed as users explore the parameters of the instrument. This work is implemented in a generic and open-source tool that enables the computation of ad hoc synthesis mappings through timbre spaces, facilitating and speeding up the workflow to get a customized sonic control system.

Keywords: interactive timbre space, generative mapping, perceptual sound synthesis.

1. Introduction

The sonic potential of synthesisers has been constantly progressing in the past decades, supported by novel techniques and rapid advances in related technologies. These devices provide almost boundless resources for the creative task of composers, designers, and musicians. Users directly interact with variables of the synthesis algorithm, as in the early days of sound synthesiser, when these were mostly operated by engineers and scientists. Generating a sound with intended timbre characteristics requires sufficient control intimacy (Fels, 2000; Moore, 1988) that implies knowledge of the relationship between parameters and sonic output, which may be complex. Most synthesis methods present a weak or missing relationship to sound perception models (Wishart, 1996). This determines a sound interaction that is object oriented rather than model oriented. Existing control strategies are typically process-based rather than result-based, which is unusual in Human Computer Interaction (HCI). This can drastically limit the efficiency of users' creative workflow. The time spent in implementing a sonic intent may exceed the time spent in conceptualizing the ideas, contradicting the creative nature of the task. Mapping strategies have contributed in easing the complexity and boosting the expressivity of the interaction (Miranda & Wanderley, 2006). The use of machine learning techniques for mapping the user input to instrument parameters has been proposed too. However, the control abstraction has not changed and the full exploitation of the synthesis potential is still challenging for users.

To address this problem, control strategies that map the user input onto synthesis variables through timbre spaces or perceptually related layers, reviewed in Section 2, have recently proliferated. Along this line, in our previous works (Fasciani, 2014; Fasciani & Wyse, 2012, 2015) we introduced a control method, implemented in an open-source software, which concurrently address the high dimensionality of synthesis control spaces and the lack of relationship between variation of control parameters and the timbre of the generated sound. In addition, our work introduced an unsupervised and automated generative mapping, independent of the synthesis algorithm and implementation, which does not require users to provide training data. Machine listening and machine learning techniques are employed to compute mappings that, given a set of synthesis variable parameters, maximize the breadth of the explorable perceptual sonic space covered by the synthesizer, and minimize the possible timbre losses due to the lower dimensionality of the control. The resulting interactive timbre space can drive simultaneously any number of parameters, and it is adapted to generic controllers, with components are directly mapped onto the most varying timbral characteristics of the sound.

A usability limitation of our system is the time required for the mapping generation. This includes the time for executing the automated parameters-to-sound analysis of the specific synthesizer (it generates the training data), and the time to compute the mapping with machine learning algorithms. The time increases exponentially with the number of synthesis parameters users aim to control through the timbre space. Here we address this issue introducing improvements in method and implementation. The adoption of the Extreme Learning Machines (ELM) algorithm (Huang, Zhu, & Siew, 2006) to train the Artificial Neural Network (ANN) performing the regression between control and timbre space, significantly reduces computation time and improves accuracy. Further reduction is achieved performing the parameters-to-sound analysis interactively as users operate the synthesizer.

The system described in this paper enables users to compute ad hoc timbre-oriented interaction, reducing the dimensionality of synthesis control space without losing timbre expressivity, which is central in modern music, despite its high dimensionality and blurry scientific definition (Risset & Wessel, 1999). The sense of hearing provides a secondary feedback on performer-instrument interaction, but for digital musical instrument this relationship can be loose. This work provides a tighter relationship between control and sound perception, independently of the specific control modality. It contributes to a reduction in the overhead for the realization of sonic intent arising by the complexity of sound generation algorithms. Further, it supports users' customization of instrument mapping while minimizing setup time and effort. **Figure 1** illustrates the traditional synthesis control method against the proposed one, which hides the technical parameters and provides higher correlation between sound and user input.

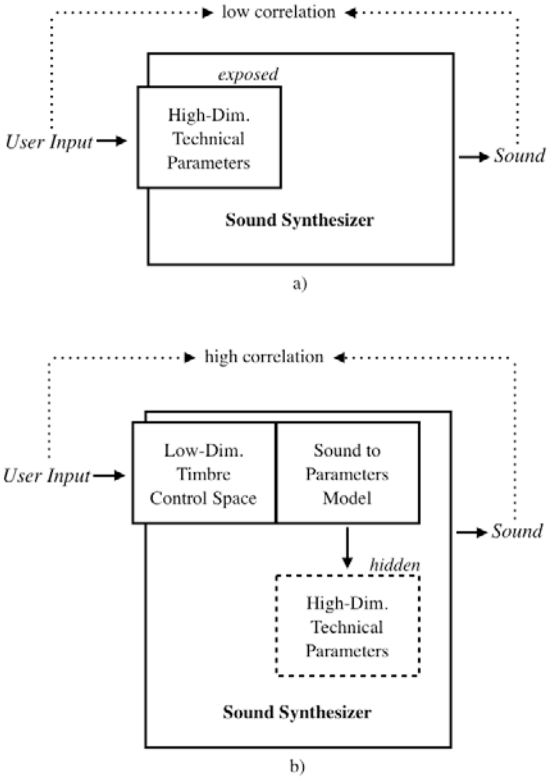

low correlation
exposed
User Input
High-Dim. Technical Parameters
Sound
Sound Synthesizer
a)
high correlation
User Input
Low-Dim. Timbre Control Space
Sound to Parameters Model
Sound
hidden
High-Dim. Technical Parameters
Sound Synthesizer
b)

Figure 1: Diagrams of a) traditional and b) proposed synthesis control method. The introduction of the timbre space hides the technical parameter and reduces the dimensionality of the control.

The rest of the paper is organized as follows. In Section 2 we present a survey on methods for perceptually related sound synthesis control. Section 3 includes a brief summary of our generative mapping strategy and sonic interaction. In Section 4 we introduce the interactive and optimized the mapping computation. Section 5 describes system integration and implementation. Conclusion and future work are discussed in Section 6.

2. Perceptually related synthesis control strategies

A synthesis control strategy is perceptually related when it explicitly manipulates timbral attributes of the generated sound. Timbre is considered "the psychoacoustician's multidimensional waste-basket category for everything that cannot be labeled pitch or loudness" (McAdams & Bergman, 1979), or the attribute that allows to distinguish two sounds with equal pitch and loudness. Timbre as a whole can be measured only at nominal level. It can be represented with a high dimensional space, whose individual components are acoustic ordinal descriptors (e.g. brightness, noisiness, color, attack, decay, vibrato, tremolo). For an accurate representation, the computation of the descriptors must include psychoacoustic principles and nonlinearities of the human auditory model. In specific applications, the timbre space can be reduced to few dimensions with multidimensional scaling techniques (Grey, 1977). However, it is necessary to preserve the element-by-element proximity between original and low-dimensional space by using nonlinear reduction techniques and non-Euclidean distances (McAdams & Cunible, 1992).

The control parameters of some synthesis techniques have a tighter relation with the output timbre due to the intrinsic characteristic of the algorithm generating sound. Additive synthesis provides control of amplitude and frequency of individual spectral components, which explicitly characterize the timbre of both harmonic and inharmonic sounds. When sound synthesis is based on a database of heterogeneous samples, it is possible to analyze the timbre characteristics of the sources and use this information in the control layer. Concatenative synthesis (Schwarz, Beller, Verbrugghe, & Britton, 2006), an extension of granular synthesis, explicitly provides control over specific sound descriptors. The synthesis sequences grains from a corpus of segmented and analyzed samples, according to the coordinate of a low dimensional interactive descriptor space, which represent an interactive timbre control structure. Through experience, humans can identify mechanical aspects (material, shape, cause, location) of the event that generates sound by auditory cues (Emmerson, 1998). In the same way we can predict the sound of a mechanical event, including performers playing musical instruments. Physical modelling synthesis simulates the mechanical phenomenon governing acoustic sound generation. The correlation between control and timbre is still complex and nonlinear, but known to users. Therefore, explicit timbre manipulation is almost immediate.

Timbre-oriented control strategies has been proposed from the pioneer work of Wessel (1979), in which subjective timbre dissimilarities between orchestral instruments synthesis patches were measured, reduced to a 2D interactive space, and then used to control the additive synthesizer. The parameters were generated by interpolating between those of the original patches used for the listening tests. This allowed generating a wider spectrum of timbres than that presented in the listening tests. Moreover this provided a drastic reduction of the control space and perceptually meaningful control dimensions. Subjective listening was replaced with computerized perceptual analysis of sounds by Jehan & Schoner (2001) in a synthesis engine that models and predicts the timbre of acoustic instrument. The analysis included in pitch, loudness, and timbre's descriptors such as brightness, noisiness and energy of the bark critical bands. Cluster-weighted modeling approximates synthesis parameters from timbre descriptions, predicts a timbre given a new set of parameters. Arfib, Couturier, Kessous, & Verfaille (2002) generalized the different approaches for a timbre-oriented control introducing perceptual layer in the mapping from gesture to musical instrument. Including the intermediate perceptual space in the modular mapping of musical interfaces improves sensitivity and efficiency of the interaction. Seago (2013) proposes a method for searches in timbre spaces based on weighted centroid localization, claiming the quasi-linearity of these spaces and explicit association of individual dimensions with acoustical attributes. Therefore, searches should return a single and optimal solution. This technique addresses usability problems of sound synthesis and it affords engagement with the sound itself, rather than with irrelevant timbre visual representations.

When the parameters-to-sound relationship of a specific synthesis algorithm is unknown, it is still possible to implement a perceptually related control by measuring psychoacoustic sound descriptors and generating a model of the synthesizer response, required for the mapping. This approach presents challenges in computing an accurate model and in retrieving coherent parameters, but it guarantees independency between control strategy and synthesis method. In this work we adopt and extend a similar strategy, which is also found in other systems. A timbre space composed of the principal components of the spectral output of an FM synthesizer is used to generate interactively auditory icons and 'earcons' (Nicol, Brewster, & Gray, 2004). A generic and modular framework, including feature evaluators, parametric synthesizers, distance metrics, and parameter optimizers controls the synthesis with a set of well-defined and arbitrary acoustic features (Hoffman & Cook, 2006), extending the work of Puckette (2004). This approach minimizes the distance between target and synthesized sonic characteristic, which are user-defined or computed from a reference sound. Klügel, Becker, & Groh (2014) propose a collaborative and perceptually related sound synthesizer controlled by multi-touch tabletop, where the interactive timbre space is mapped and visually represented. Generative topographic techniques are used to map perceptual audio features to synthesis parameters. Computational challenges in performing a deep time-frequency analysis with high dimensional parameters space are identified but not addressed. An approach based on fuzzy models, manually defined by experts, expose to users a large set of intuitive and natural timbre descriptors, which allows novice users to effectively control the synthesis algorithm by visual programming (Pošćić & Kreković, 2013).

Strategies for perceptually related control of audio mosaicing systems has been introduced as well. In this case the sound synthesis is replaced with the sequential playback of samples retrieved from a database organized according to audio descriptors. This generates evolving timbre and textures. Proposed implementations (Lazier & Cook, 2003; Schnell, Cifuentes, & Lambert, 2010; Grill, 2012) map the analyzed samples collection onto a two- or three-dimensional sonic map defined after long- and short-term sound analyses.

3. Timbre space for synthesis control

This section briefly summarizes our approach to perceptually related synthesis control through timbre spaces. We assume having no prior knowledge of the synthesis algorithm and of the relationship between parameters and output sound, neither this provided by users. The parameters-to-sound model is derived from input-output observations. In particular we use machine listening techniques to 'hear' the timbre almost like humans. Further we assume univocal relation between parameters and sound, excluding random variables in the generation algorithm, while we tolerate not univocal sound to parameters relationship. Therefore, this control strategy is compatible with any deterministic sound synthesizer.

Performing a full parameters-to-sound characterization of a synthesizer is a demanding task. The set of possible parameter permutations we have to analyze grows exponentially with the number of parameters and their possible values, which is equal to at least 127 for the real-valued ones (e.g. 33G permutations with 5 real-valued parameters). The full characterization requires excessive computational resources and time. In this context this is generally not required. Firstly, when performing, users interact only with few synthesis parameters, the remaining are fixed given a specific synthesis patch. Secondly, the control resolution of real-valued parameters is high to avoid sound glitches in the real-time synthesis control, though for the analysis, a lower resolution can still provide accurate modeling while drastically reducing the number of permutations to measure. Further, the sound-to-parameters analysis aims to identify only the timbre components that change varying the non-fixed synthesis parameters. We do not intend to carry out a comprehensive timbre characterization. Therefore, users can contribute in reducing analysis and mapping complexities also by discarding descriptors that are steady or not relevant, and by limiting the analysis only to a specific sub-region of the ADSR envelope. Further, the method presented here is independent of the specific descriptors selected for the analysis, detailed in Section 5.

Given the set of variable parameters, with their analysis range and resolution, we compute the matrix **I** which includes all permutation vectors $\mathbf{i}_j$ of variable parameters. One at a time and automatically, the synthesizer is driven with the $\mathbf{i}_j$, and the sound is analyzed computing vectors of perceptually related descriptors $\mathbf{d}_j$, accumulated in **D**. The matrices **I** and **D** model the parameters-to-sound relationship of a specific synthesizer, according to user-defined settings. The matrices have different dimensionality but identical number of elements ($\mathbf{i}_j$ and $\mathbf{d}_j$, associated pairwise).

Our control strategy maps the user control space **C** onto the parameter space **I** through the timbre space **D**. We assume that **C** has up to three components, which are statistically independent and uniformly distributed in a fixed range. Most general-purpose musical controllers are compliant with this model. For instance the control data generated interacting with a touchpad can be represented with a uniformly distributed square. The proposed generative mapping strategy address issues such as high dimensionality of the timbre space **D**, the arbitrary shape and distribution of **D**, and possible not one-to-one relationship between vectors $\mathbf{d}_j$ and $\mathbf{i}_j$ (similar timbres generated with different synthesis parameters). To compute the mapping, the dimensionality of **D** is reduced to the one of **C** using ISOMAP (Tenenbaum, Silva, & Langford, 2000). Then, the entries in the low-dimensional timbre space $\mathbf{D}^*$ are rearranged into a uniformly distributed space $\mathbf{D}_U^*$ (line, square or cube) using a homotopic iterative transformation that preserve the neighbourhood relationships (Nguyen, Burkardt, Gunzburger, Ju, & Saka, 2009). The obtained space $\mathbf{D}_U^*$ shares identical geometrical and statistical characteristics with the space **C**. We model the inverse of this transformation with the function *m( )*, which is implemented with a feed-forward ANN performing the regression from the low-dimensional and uniformly distributed timbre space $\mathbf{D}_U^*$ to the low-dimensional timbre space $\mathbf{D}^*$. The ANN is trained with a back propagation iterative algorithm. Reduction and redistribution do not alter the number of entries in the timbre space. Pairwise association between $\mathbf{d}^*_j$ or $\mathbf{d}_U{}^*_j$ and the relative $\mathbf{i}_j$ are still valid. The mapping from the user control vector **c** to the synthesis parameter vector **i** is therefore defined in the equation below.

$$\mathbf{i} = \mathbf{i}_j : \underset{j}{\operatorname{argmin}} \left| \mathbf{d}^*_j - m(\mathbf{c}) \right|$$

The mapping returns the parameter vector **i** equal to the $\mathbf{i}_j$ of **I** associated with the timbre vector $\mathbf{d}^*_j$ in $\mathbf{D}^*$ closer to the projection of the user control vector **c** onto the reduced timbre space *m(***c***)*. To cope with the limited parameter resolution of the analysis stage, the real-time mapping includes temporal and spatial interpolations (inverse distance weighting) of the synthesis parameters, as detailed in (Fasciani, 2014). For addressing the possible parameter discontinuities in the $\mathbf{i}_j$ stream due to the not one-to-one relationship between sound and parameters, we also presented a search strategy for the closer $\mathbf{d}^*_j$ restricted to the subset of $\mathbf{D}^*$ that guarantee continuity and space explorability.

In **Figure 2** we illustrate a basic example of our control strategy, which provides a perceptually related and dimensionality reduced synthesis control space. The figure shows a two-dimensional parameter space. Each parameter can assume 8 different values, giving a total 64 permutations, represented with the black circles. The blue, green, and red lines illustrate different divergent mappings that reduce the control space from two to one dimension. The linear mapping, in blue, can retrieve only few entries in the parameter space, losing most permutations and likely timbres. The nonlinear mapping, in green, arranges the 64 possible parameter permutations on a single line, but it does not guarantee any timbre continuity or meaningful organization of the generated sound. Finally the red dotted line represents our mapping strategy, where the 64 possible synthesizer states are sorted on a single dimension according to their previously analysed timbre. The arrangement can follow the most varying perceptually related descriptor within the parameter space, or a user-defined descriptor. This mapping appears random in the parameter space but it is systematic in the timbre space. This control strategy supports any dimensionality of the parameter space **I**, timbre space **D**, and user control space **C**.

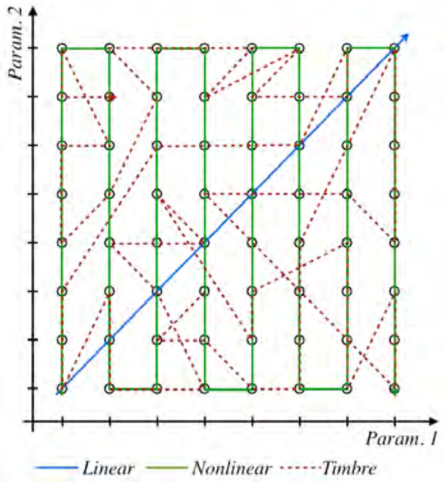


Figure 2: Illustration of synthesis control space two-to-one reduction, using linear mapping (blue), nonlinear mapping (green), and timbre space mapping (red dotted). For the latter, the synthesis states are sorted on a single control dimension according to timbre characteristics.

4. Interactive and optimized timbre space computation

Implementing the method described above, the time required for performing the parameters-to-sound analysis and computing the mapping was often exceeding half an hour. The time strongly depends on the number of entries in **I** and on the dimensionality of **C**. We address this issue introducing an interactive analysis and mapping strategy improvements.

To speed up the analysis, the execution can be run in real-time while users interactively explore the parameters space of the synthesizer. Parameters and descriptor vectors are progressively accumulated in **I** and **D**. The analysis can be executed on the current parameter configuration generating a single pair **i**–**d**, or it can run continuously producing a stream of **i** and **d**. After setting the synthesis parameters to **i** we typically wait 50-100 ms before starting to analyze the sound. This allows sufficient time to the synthesizer for responding to the new configuration. This is not possible when vectors **i** are presented as a stream and the analysis runs continuously. To compute a reliable parameters-to-sound model it is therefore essential to measure and consider the response time of the synthesizer. The two streams of vectors have to be re-aligned associating each **i** with the corresponding **d**. In our implementation, presented in Section 5, Virtual Studio Technology (VST) synthesizers are hosted in Max/MSP. In this context we found that the **i** to **d** misalignment T_{resp} (measured in number of vectors) depends on analysis window size, analysis window step, and Max/MSP signal vector size, as in the equation below.

$$T_{resp} = \mathrm{round}\Big((win\,/\,step) + (vect_{MaxMSP}\,/\,step)\Big)$$

When performing the analysis we usually compute multiple **d** for each **i**, which are post-processed and merged into a single descriptor vector. This reduces measurement noise taking the average of more observations, and allows including longer-term features of instantaneous descriptors, such as their variation range and periodicity. In the continuous analysis this can be performed in a later stage, by grouping descriptor vectors **d** related to the same parameter vector **i**. However, it is unlikely to find identical vectors **i** when users interact with high-resolution controllers. This is detrimental for the system because **I** and **D** would determine poor modeling (a single observation **d**) and high computational load (large number of **d**-**i** pairs). Hence, before computing the mapping we reduce the resolution of the parameters in **I** to user-defined values. Then we merge **i**-**d** pairs with identical synthesis parameters, computing average and range of the descriptors. The periodicity cannot be estimated because the observations may not be consecutive. We also provide an alternative method to reduce the automatic analysis time, running VST and Max/MSP in non-real-time mode so that audio samples are generated and analyzed at a rate higher than the audio sampling period, fully using the CPU power.

In the mapping computation the most computationally demanding part was the back propagation iterative training algorithm for the multilayer ANN, which performs the regression $\mathbf{D}_U^*$ to $\mathbf{D}^*$. We replaced this with the ELM (Huang et al., 2006), which is an efficient and not iterative training procedure. Conventional learning methods require seeing the training data before generating parameters of the hidden neurons, while ELM can randomly generate these in advance. With ELM we train a single-hidden layer feed-forward ANN randomly choosing the input weights and bias coefficients of the hidden nodes. Thereafter the output weights are computed analytically. The introduction of ELM allows the training in shorter time of a larger ANN, which determines faster mapping computation and higher accuracy. Over a large set of mapping test cases, the introduction of ELM training algorithm determined an average 98% reduction of the training time, a 87% reduction of the real-time mapping time, and a sensible improvement of the regression accuracy, measured with the Mean Squared Error (MSE). We use a growing algorithm to design the ANN providing a satisfactory regression. Starting with 10 neurons in the hidden layer, we grow the network by 10 units at a time, allowing a maximum of 200 neurons, until the MSE is below 0.01.

The homotopic uniform redistribution and of the ISOMAP dimensionality reduction are other computationally demanding components. In the functional prototype, described in Section 5, their implementation has been optimized to provide a tradeoff between accuracy and fast execution. Moreover we integrated synthesizer, analysis, mapping computation, real-time mapping, visualizations, and user interface in a single environment. This had improved the computation efficiency, reduced the data exchange overhead, and simplified the user workflow, executed in a single Graphical User Interface (GUI). Overall we reduced by 82% the mapping computation time, and by 94% the analysis automated computation time. The alternative interactive analysis determines no time overhead for computing the interactive timbre space. In **Figure 3** we illustrate the previous and current system implementations. The first was also part of a system for the vocal control of sound synthesis through timbre spaces. The seamless integration of the different components of the system further simplifies the procedure to compute personalized mappings through the timbre space, according to users' preferences and customized for their specific synthesizers.

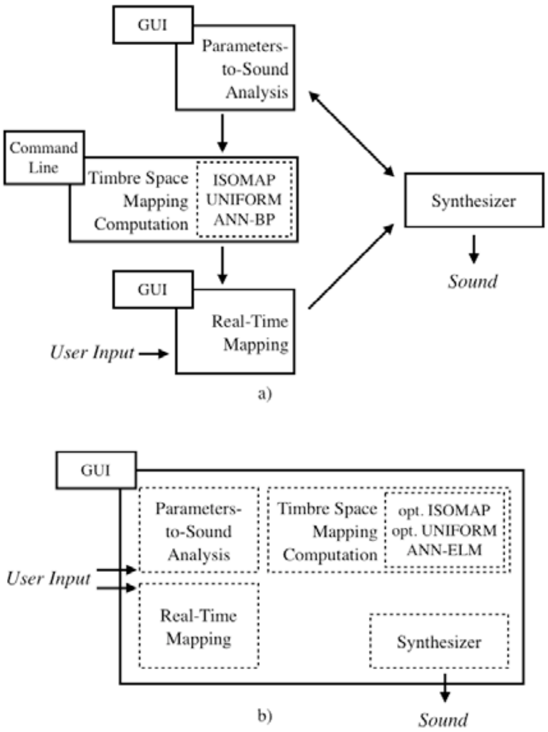
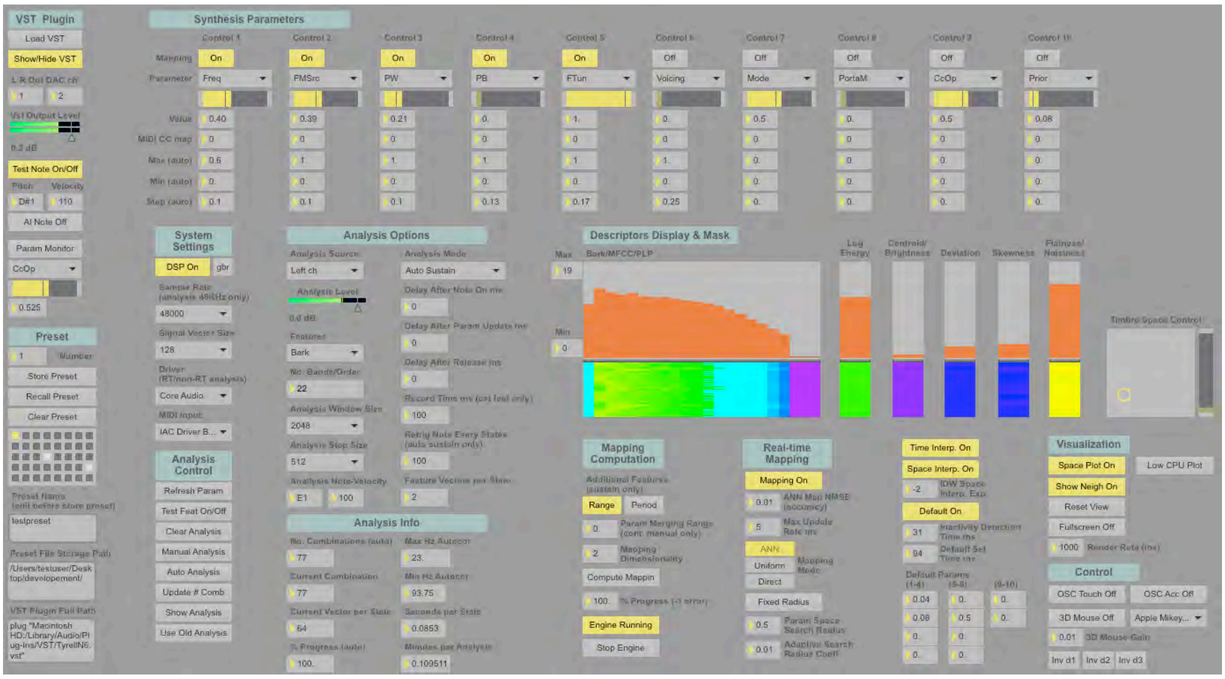


Figure 3: Illustration of a) previous and b) current system implementations. The latter, including improved mapping algorithms, presents a single GUI and seamless integration of the different components.

The proposed timbre space mapping method provided satisfactory performances (evaluated in Fasciani, 2014; Fasciani & Wyse, 2015). We measured quantitative metrics such as the percentage of retrievable synthesis parameter permutations, parameter continuity, and timbre space losses over an extensive set of synthesizers. The computational optimizations and usability improvements described here did not determine any performance drop.

5. Fully-functional integrated prototype

The sound-to-parameter analysis, the mapping computation, and the real-time mapping have been implemented in a standalone and fully functional prototype, presenting an integrated GUI, which exposes system settings and mapping options. Prototype, source code, user guide, screenshots, and demo videos are available at http://stefanofasciani.com/tsm4vsts.html. The prototype is implemented in Max/MPS and it hosts VST software synthesizer using the native vst~ object. The FTM library (Schnell, Borghesi, Schwarz, Bevilacqua, & Muller, 2005) and ad-hoc externals are extensively used for the analysis and real-time mapping. The ISOMAP and homotopic uniform redistribution algorithms are particularly complex and have not yet been ported to Max/MSP. Therefore a part of the mapping computation is still implemented in a MATLAB script. However this has been compiled in a standalone executable that runs in background and communicates with Max/MSP via the Open Sound Control (OSC) protocol. This process is completely transparent to users. This system can be seen as a VST synthesizers wrapper that exposes an alternative control strategy, as illustrated in the diagram of **Figure 3** b). The VST and synthesis patch can be stored to and recalled from presets, which also save mapping, analysis and system settings. The GUI of the prototype, in **Figure 4**, includes advanced options to further customize the timbre space mapping. The system implementation provides sufficient flexibility to support different workflows and interaction strategies. The parameters-to-sound analysis includes five different modalities. Available analysis descriptors include loudness of Bark critical bands, MFCC, PLP, and spectral moments, with possible addition of variation range and periodicity. The mapping can be recomputed changing most timbre options without executing a new analysis. Users are provided with options to tune the mapping in real-time. An interactive color-coded visualization of the timbre space, as in **Figure 5**, supports the development of control intimacy with mappings generated unsupervisedly.



VST Plugin

Load VST

Show/Hide VST

L R Out DAC ch: 1 2

Vst Output Level: 0.3 dB

Test Note On/Off

Pitch: D#1 Velocity: 110

Al Note Off

Param Monitor: CcOp 0.525

Preset

Number: 1

Store Preset

Recall Preset

Clear Preset

Preset Name (edit before store preset): testpreset

Preset File Storage Path: /Users/testuser/Desktop/developement/

VST Plugin Full Path: plug "Macintosh HD:/Library/Audio/Plug-Ins/VST/TyrellN6.vst"

Synthesis Parameters

	Control 1	Control 2	Control 3	Control 4	Control 5	Control 6	Control 7	Control 8	Control 9	Control 10
Mapping	On	On	On	On	On	Off	Off	Off	Off	Off
Parameter	Freq	FMSrc	PW	PB	FTun	Voicing	Mode	PortaM	CcOp	Prior
Value	0.40	0.39	0.21	0.	1.	0.	0.5	0.	0.5	0.08
MIDI CC map	0	0	0	0	0	0	0	0	0	0
Max (auto)	0.6	1.	1.	1.	1.	1.	0.	0.	0.	0.
Min (auto)	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
Step (auto)	0.1	0.1	0.1	0.13	0.17	0.25	0.	0.	0.	0.

System Settings

DSP On gbr

Sample Rate (analysis 48KHz only): 48000

Signal Vector Size: 128

Driver (RT/non-RT analysis): Core Audio

MIDI input: IAC Driver B...

Analysis Control

Refresh Param

Test Feat On/Off

Clear Analysis

Manual Analysis

Auto Analysis

Update # Comb

Show Analysis

Use Old Analysis

Analysis Options

Analysis Source: Left ch

Analysis Level: 0.0 dB

Features: Bark

No. Bands/Order: 22

Analysis Window Size: 2048

Analysis Step Size: 512

Analysis Note-Velocity: E1 100

Analysis Mode: Auto Sustain

Delay After Note On ms: 0

Delay After Param Update ms: 0

Delay After Release ms: 0

Record Time ms (cal feat only): 100

Retrig Note Every States (auto sustain only): 100

Feature Vectors per State: 2

Analysis Info

No. Combinations (auto): 77

Current Combination: 77

Current Vector per State: 64

% Progress (auto): 100.

Max Hz Autocor: 23.

Min Hz Autocor: 93.75

Seconds per State: 0.0853

Minutes per Analysis: 0.109511

Descriptors Display & Mask

Max: 19

Min: 0

Mapping Computation

Additional Features (sustain only): Range Period

0. Param Merging Range (cont. manual only)

2 Mapping Dimensionality

Compute Mappin

100. % Progress (-1 error)

Engine Running

Stop Engine

Real-time Mapping

Mapping On

0.01 ANN Map NMSE (accuracy)

5 Max Update Rate ms

Mapping Mode: ANN Uniform Direct

Fixed Radius

0.5 Param Space Search Radius

0.01 Adaptive Search Radius Coeff

Time Interp. On

Space Interp. On

-2 IDW Space Interp. Exp

Default On

31 Inactivity Detection Time ms

94 Default Set Time ms

Default Params (1-4)	(5-8)	(9-10)
0.04	0.	0.
0.08	0.5	0.
0.	0.	
0.	0.	

Visualization

Space Plot On

Low CPU Plot

Show Neigh On

Reset View

Fullscreen Off

1000 Render Rate (ms)

Control

OSC Touch Off

OSC Acc Off

3D Mouse Off

Apple Mikey...

0.01 3D Mouse Gain

Inv d1 Inv d2 Inv d3

Figure 4: GUI of the fully functional prototype implementing the timbre space mapping for VST synthesizers.

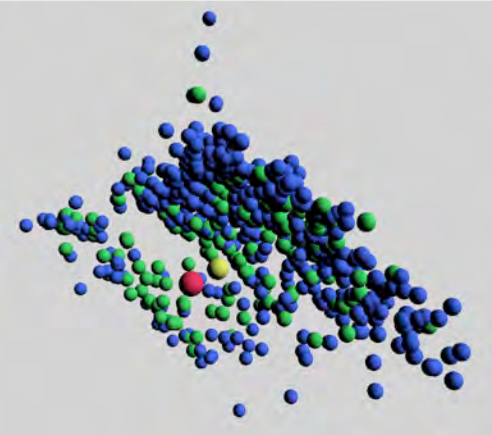


Figure 5: 3D interactive timbre space visualization. The blue spheres represent the entries in the reduced timbre space $\mathbf{D}^*$, the red one is the projection *m(***c***)* of the user control with the yellow one being the closer $\mathbf{d}^*_j$. The green spheres represent the instantaneous restricted search space. Camera position and angle are user-defined.

6. Conclusion and future work

We have presented a method to interactively compute timbre space of specific sound synthesizers, and to use this as control structure, which provides a perceptually related interaction. Our previous studies on a system presenting a similar interaction approach, demonstrated that the control through the timbre space successfully hides the synthesis technical parameters, reduces the control space, and provides perceptual related interaction. However, further comprehensive user-studies are necessary to evaluate the interactive and unsupervised timbre mapping generation. The implementation of an open-source and fully functional prototype, compatible with any software synthesiser, can be a resource for performers and musicians. At the same time, it can be used as an exploration and tool by researchers and developers to study alternatives and improvement to the proposed method, including the use of different sound descriptors. The algorithm and the implementation can be further improved finding suitable alternatives to the ISOMAP and homotopic uniform redistribution algorithms, or introducing drastic optimizations. These can be ported in Max/MSP for the system integration in a single programming environment.

References

Arfib, D., Couturier, J. M., Kessous, L., & Verfaille, V. (2002). Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces. Organized Sound, 7(2), 127–144.

Emmerson, S. (1998). Aural landscape: musical space. Organised Sound, 3(2), 135–140.

Fasciani, S. (2014). Voice-controlled interface for digital musical instruments (Ph.D. Thesis). National University of Singapore, Singapore.

Fasciani, S., & Wyse, L. (2012). Adapting general purpose interfaces to synthesis engines using unsupervised dimensionality reduction techniques and inverse mapping from features to parameters. In Proceedings of the 2012 International Computer Music Conference. Ljubljana, Slovenia.

Fasciani, S., & Wyse, L. (2015). Vocal control of digital musical instruments personalized by unsupervised machine listening and learning. Submitted, ACM Transactions on Intelligent Systems and Technology.

Fels, S. (2000). Intimacy and embodiment: implications for art and technology. In Proceedings of the 2000 ACM workshops on Multimedia (pp. 13–16).

Grey, J. M. (1977). Multidimensional perceptual scaling of musical timbres. Journal of the Acoustical Society of America, 61(5), 1270–1277.

Grill, T. (2012). Constructing high-level perceptual audio descriptors for textural sounds. In Proceedings of the 9th Sound and Music Computing international conference. Copenhagen, Denmark.

Hoffman, M., & Cook, P. R. (2006). Feature-based synthesis: Mapping acoustic and perceptual features onto synthesis parameters. In Proceedings of the 2006 International Computer Music Conference. New Orleans , US.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1–3), 489–501.

Jehan, T., & Schoner, B. (2001). An audio-driven perceptually meaningful timbre synthesizer. In Proceedings of the 2001 International Computer Music Conference. Havana, Cuba.

Klügel, N., Becker, T., & Groh, G. (2014). Designing Sound Collaboratively Perceptually Motivated Audio Synthesis. In Proceedings of the 14th International Conference on New Interfaces for Musical Expression. London, United Kingdom.

Lazier, A., & Cook, P. R. (2003). Mosievius: feature driven interactive audio mosaicing. In Proceedings of the 7th international conference on Digital Audio Effects. Napoli, Italy.

McAdams, S., & Bergman, A. (1979). Hearing musical streams. Computer Music Journal, 3(4), 26–43, 60.

McAdams, S., & Cunible, J. C. (1992). Perception of timbral analogies. Royal Society of London Philosophical Transactions, 336, 383–389.

Miranda, E. R., & Wanderley, M. M. (2006). New digital musical instruments: control and interaction beyond the keyboard. A-R Editions, Inc.

Moore, F. R. (1988). The dysfunctions of MIDI. Computer Music Journal, 12(1), 19–28.

Nguyen, H., Burkardt, J., Gunzburger, M., Ju, L., & Saka, Y. (2009). Constrained CVT meshes and a comparison of triangular mesh generators. Computational Geometry, 42(1), 1–19.

Nicol, C., Brewster, S. A., & Gray, P. D. (2004). Designing Sound: Towards a System for Designing Audio Interfaces using Timbre Spaces. In Proceedings of the 10th International Conference on Auditory Display. Sydney, Australia.

Pošćić, A., & Kreković, G. (2013). Controlling a sound synthesizer using timbral attributes. In Proceedings of the 10th Sound and Music Computing international conference. Stockholm, Sweden.

Puckette, M. (2004). Low-dimensional parameter mapping using spectral envelopes. In Proceedings of the 2004 International Computer Music Conference. Miami, US.

Risset, J. C., & Wessel, D. (1999). Exploration of timbre by analysis and synthesis. The Psychology of Music, 113–169.

Schnell, N., Borghesi, R., Schwarz, D., Bevilacqua, F., & Muller, R. (2005). FTM - Complex Data Structure for Max. In Proceedings of the 2005 International Computer Music Conference. Barcelona, Spain.

Schnell, N., Cifuentes, M. A. S., & Lambert, J. P. (2010). First steps in relaxed real-time typo-morphological audio analysis/synthesis. In Proceeding of the 7th Sound and Music Computing international conference. Barcelona, Spain.

Schwarz, D., Beller, G., Verbrugghe, B., & Britton, S. (2006). Real-time corpus-based concatenative synthesis with CATART. In Proceedings of the 9th international conference on Digital Audio Effects (pp. 279–282). Montreal, Canada.

Seago, A. (2013). A New Interaction Strategy for Musical Timbre Design. In S. Holland, K. Wilkie, P. Mulholland, & A. Seago (Eds.), Music and Human-Computer Interaction (pp. 153–169). Springer.

Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500), 2319.

Wessel, D. (1979). Timbre space as a musical control structure. Computer Music Journal, 3(2), 45–52.

Wishart, T. (1996). On Sonic Art. Harwood Academic Publishers.