Vocal Control of Sound Synthesis Personalized by Unsupervised Machine Listening and Learning

Stefano Fasciani* and Lonce Wyse†

*Faculty of Engineering and Information Sciences
University of Wollongong in Dubai
Knowledge Park, Block 15, P.O. Box 20183
Dubai, United Arab Emirates
stefanofasciani@stefanofasciani.com

†Communication and New Media Department
National University of Singapore
11 Computing Drive, 117416, Singapore
lonce.wyse@nus.edu.sg

Abstract: In this article we describe a user-driven adaptive method to control the sonic response of digital musical instruments using information extracted from the timbre of the human voice. The mapping between heterogeneous attributes of the input and output timbres is determined from data collected through machine-listening techniques and then processed by unsupervised machine-learning algorithms. This approach is based on a minimum-loss mapping that hides any synthesizer-specific parameters and that maps the vocal interaction directly to perceptual characteristics of the generated sound. The mapping adapts to the dynamics detected in the voice and maximizes the timbral space covered by the sound synthesizer. The strategies for mapping vocal control to perceptual timbral features and for automating the customization of vocal interfaces for different users and synthesizers, in general, are evaluated through a variety of qualitative and quantitative methods.

The importance of mapping in the design of expressive interfaces for musical instruments has long been recognized (Poupyrev et al. 2001; Hunt, Wanderley, and Paradis 2003). Mapping techniques have been at the heart of the development of novel musical instruments (Wanderley and Depalle 2004) and of generic interactive sound controllers. Regardless of the input technology used, most interfaces for digital musical instruments require physical motor interaction from users, captured with or without contact (Bongers 2000). A large proportion of interfaces are built for the hands, evinced by a focus of many devices on tactile, haptic, and gestural input modalities. Human physical limits, however, restrict the number of parameters or events that performers can control simultaneously using these. Few-to-many mappings or preprogrammed automations are common workarounds. With the exception of “augmented instruments” that aim to saturate performers’ spare bandwidth (Cook 2001), this issue is rarely addressed in the design phase of instrument interfaces.

The aim of the framework described in this article is to provide users with a simple system to implement personalized vocal control of sound generators, adapted to the sonic characteristics both of their voice and of the output of the specific synthesis process of their choice. This enables the augmentation of control dimension through voice for controlling synthesis algorithms. The system is largely self-configurable, minimizing user effort in setting up the personalized interface. The balance between simplicity of user requirements and richness of musical expression (Stolterman 2008) is achieved in two stages. First, we establish a natural interaction between voice and sound synthesizers, implementing the mapping on the perceptually related timbre layer. Second, we develop and combine techniques of machine listening and of unsupervised learning to compute the mapping automatically from a minimal set of user-provided information. The framework we describe here addresses [1] expansive and complex timbral capabilities of sound synthesis algorithms, which require high-dimensional non-linear control spaces; [2] noisy characteristics of the vocal control signal, which challenge unsupervised mapping creation; and [3] interaction with perceptually relevant characteristics of the sound rather than with the complicated synthesis parameters. The resulting
mapping strategy maximizes the coverage of the perceptual sonic space of synthesizers given the span of the controlling voice's timbral domain. It also minimizes the cognitive complexity of controlling sounds embedded in high-dimensional parameter spaces by providing a lower-dimensional control with unchanged timbral potential. This article summarizes the system integrating these components, some of which we have described in more detail elsewhere (Fasciani and Wyse 2013; Fasciani 2016), and then evaluates the effectiveness of automated vocal-to-synthesizer timbre mapping strategies through a variety of qualitative and quantitative user studies.

Related Work

Indirect gestural acquisition refers to strategies to extract control data from a sound source (Wanderley and Depalle 2004). Puckette and Lippe (1994) proposed the extraction of control intention from the sound of a clarinet by tracking the temporal dynamics of amplitude, pitch, and the principal components of timbre descriptors. Analysis of the timbre of plucked guitar based on mel-frequency cepstral coefficients (MFCCs), psychoacoustic parameters, time–frequency analysis, principal component analysis, and estimation of plucking point has been used to implement musical interaction (cf. Orio 1999, Traube, Depalle, and Wanderley 2003). In contrast to synthesis algorithms, acoustic instruments generally allow only minor timbral nuances, because the sound generated is constrained by the instruments' physical bodies. So the data extracted from these sources is not optimally suited for the control of complex synthesis engines. Generic methods to compute control data from audio signals have also been proposed (Lazier and Cook 2003; Poepel and Dannenberg 2005; Schnell, Cifuentes, and Lambert 2010). These methods rely on the extraction of fixed features such MFCCs, pitch, harmonic tracking, and timbre dynamics.

Several previous works proposed the voice as a gestural control source, owing to its wide timbral range. Oliver, Yu, and Metois (1997) extract ten dynamic parameters, such as pitch, loudness, formants, cepstra, and their deviations, that were used to interact with an ensemble of instruments. Jordi Janer (2008) developed a system to interact vocally with the synthetic emulation of acoustic instruments, singing-voice synthesis, and audio mosaicking by explicitly mapping from detected vocal pitch, energy envelope, onset, and temporal segmentation of syllables. Amaury Hazan (2005) used a set of spectral vocal features and an energy measure computed over the onset frame to query a decision tree that triggered sampled drum sounds. Kapur, Benning, and Tzanetakis (2004) applied a similar method to retrieving drum loops from a database. Teglbjærg, Andersen, and Serafin (2015) used a live voice signal instead of white noise as the excitation signal in the Karplus-Strong physical model of plucked strings. Sven König used acoustic similarities between voice and sound segments in his interactive system scrambled?HaCkZ! [a demonstration can be seen at http://youtu.be/eRLhKaxcKpA]. Other approaches use, as control signals, the principal components derived from linear predictive coding coefficients or from MFCC representations of the voice (Orio 1997; Ramakrishnan, Freeman, and Varnik 2004; Janer and De Boer 2008). Loscos and Aussenac (2005) propose the use of the area of the weighted low-frequency spectrum computed from the voice signal to control a wah-wah effect, whereas Dan Stowell (2010) computes a mapping based on the principal components of larger sets of vocal features to interact with the parameters of synthesis engines through an autoassociative regression tree. Low-level features extracted from vocal imitation of sounds have also been used for searching a collection of synthesis preset patches with reinforcement learning (Cartwright and Pardo 2014) and to search a large database of prerecorded sounds (Roma and Serra 2015). These works consider the voice as an immediate and natural means of expressing sound, and thereby enable fast “prototyping” or “sketching” of sounds (Cartwright and Pardo 2015; Delle Monache et al. 2015). The innate human ability to control the vocal timbre, rather than specific articulators, is exploited by Lemaitre and colleagues (2013), and it does not require the mastery of new skills. Most of the existing methods are designed to address specific synthesis methods or applications.
Machine listening and machine learning have been broadly used in creative applications such as sonic interactive systems and mapping for novel musical interfaces. Following the early work by David Wessel (1991) with artificial neural networks (ANNs), a variety of machine-learning techniques have been integrated in such systems, including classification, clustering, and regression algorithms (Caramiaux and Tanaka 2013). In particular, these are used for mapping purposes and for gesture recognition. Generally, musical interfaces and sonic interactive systems based on machine learning do not provide a ready-to-use mapping, but they support the implementation of personalized systems. Supervised machine learning for sonic interaction has been proposed by Bevilacqua and coworkers (2010), where a hidden Markov model is used to follow the temporal unfolding of user gestures; by Fiebrink (2011) and by Gillian, Knapp, and O’Modhrain (2011) to provide an ANN and classifiers integrated in a user-friendly generic mapping environment; and by Scurto and Fiebrink (2016) to create mappings based only on user demonstrations of control gestures. Unsupervised machine learning has been proposed as well (Smith and Garnett 2011, 2012), including the self-organizing maps (SOM) method, which has been used for automatic organization of sonic data (Ness and Tzanetakis 2009). Instrument designers aiming for generative mappings can rely on toolkits for visual programming languages to ease the integration of a wide spectrum of machine-learning algorithms (Bullock and Momeni 2015).

In this article we describe and evaluate the integration of a user-driven generative mapping framework based on several techniques we introduced in earlier work (Fasciani 2012, 2016; Fasciani and Wyse 2013). The method is independent of the specific synthesis method and it measures the perceptual timbre response of any deterministic sound synthesizer, providing low-dimensional and perceptually based interaction independent of the type and number of synthesis parameters controlled. The strategy for indirect gestural acquisition from voice includes the automated selection of features to extract and the adaptation to the users’ vocal interaction style, while providing multiple degrees of freedom. This approach depends upon machine-listening and unsupervised-learning techniques that work with minimal amounts of training data and expose a manageable number of meaningful choices to users.

**System Architecture**

There are two major components comprising the integrated system: the vocal gestural controller (Fasciani and Wyse 2013), and the synthesis timbre-space mapping (Fasciani 2016), as illustrated in Figure 1. The first is built upon robust and noise-free control signals extracted from the voice (Fasciani 2012), representative of control intention expressed by subverbal vocal gestures. The second component maps the control signals computed from the voice onto the synthesis timbre space, from which we can retrieve the parameters to control the synthesizer itself. Both components use machine listening—the first to analyze the user’s voice and the second to analyze synthesis output. The analysis data are then used to train the system that implements the voice-to-synthesis parameter mapping. We assume that the deterministic synthesis algorithm may generate any sound, whereas the sonic space of vocal articulation is limited. The voice can be noisy and inaccurate because it is generated by a human, whereas deterministic synthesis algorithms always produce the same timbre for a given set of parameter values. To provide tight coupling (i.e., low latency) between control action and system response, we consider only the instantaneous characteristics of the voice. On the other hand, in the offline analysis of the synthesizer, we can also examine the sonic variation due to parameter changes over longer periods, allowing us to include cases such as low-rate timbre modulations. Therefore, the features we compute from voice and synthesis signals, and the processing methods we use to extract relevant information, differ significantly across the two components of the system. The system architecture is based on the concept of remapping across these heterogeneous domains. The spatial representations over which both the voice and the synthesis timbres move are considered as manifolds, each with an individual shape and distribution. The generative
mapping finds the homomorphic (i.e., structure preserving) transformation that maximizes their overlap. The system is constrained to only two or three intermediate dimensions across these spaces, to limit the complexity of both computation and cognitive interaction. This also simplifies the display of user vocal interactions, an important feedback element in the user interface.

**Computation of a Control Space from Voice Data**

Using the voice as a source of musical control offers two key benefits. First, it is often “spare bandwidth” otherwise unused by performers engaged with musical interfaces. Second, it offers an immediate interaction modality with low cognitive complexity, which does not require mastering new skills. The voice, at the subverbal level, offers the kind of instantaneous, continuous, and multidimensional control essential for real-time interactive synthesis algorithms.

Togneri, Alder, and Attikiouzel (1992) showed that articulations in speech can be represented with as few as four dimensions nonlinearly embedded in higher-dimensional spaces of heterogeneous low-level features. The embedding depends on individual vocal tracts and on the specific articulations, and parameter extraction is hampered by intrinsic noise sources. The speech space is a subset of a wider vocal articulation space, but most individuals use the vocal apparatus primarily for the purpose of speech. Micromodulations are always present in human voice, even when producing steady timbres with invariant articulation (Quatieri 2008). Moreover, repeated instances of perceptually identical vocal sound can be acoustically quite different [Stevens 1971; Yang, Millar, and MacLeod 1996]. Finally, linear variations of the vocal tract articulations determine nonlinear dynamics in the acoustic features of the vocal sound. The model we learn from user-provided examples considers the inaccuracy that human-generated data can present at interaction time.

The system does not require any specific vocal sounds to operate, but it adapts to the vocal control style of users. To provide natural and intuitive interaction, the control strategy relies on the following principle: The synthesis timbre changes if and only if the instantaneous acoustic characteristics of the voice change. We define two categories of control sound, **vocal gesture** and **vocal posture**, which are sufficient to describe the interaction for the purpose of training. A vocal gesture is a sound with dynamic (i.e., variable) acoustic characteristics intended to determine a variation of the synthesis timbre. A vocal posture is a sound with steady acoustic characteristics that leaves the synthesis timbre unchanged.
Silence is a special case of posture. Vocal postures are intrinsically restricted to “continuant” articulations (those that can be sustained over time) such as vowels and nasal, liquid, or affricate consonants. The training data includes both vocal postures and vocal gestures. In the training procedure the temporal unfolding of the vocal gestures is disregarded, so vocal sounds can be presented in different orders across training instances.

**Selection of Optimal Low-Level Acoustic Features**

The next task is to identify which vocal features will be most effective for a given user and the particular gestures they plan to use to define the control they desire. The overall strategy is to start with a large number of features, including linear predictive coding coefficients, MFCCs, and perceptual linear predictive coefficients, under a variety of different computational conditions (i.e., sampling rate, window sizes, overlap, and number of coefficients), and discover which of these features is optimal. By “optimal” we mean meeting multiple constraints, the two most important being insensitivity to noise and the ability to capture the majority of the variation in the signal. Interface usability depends on the stability of the computed features. Vocal postures, intended to be static, contain short-term noise-like variations in the low-level features, which might be propagated to the synthesizer. Additionally, features computed from vocal gestures and used for mapping purposes can be redundant, which could undermine the “browsability” of the timbre space. Therefore, instead of choosing a fixed set of low-level features, as in most related work, we observe the statistical properties of a large set of features across the training data and then identify those features and their computational conditions that best represent the vocal data.

To determine the feature set, for each vocal posture we compute the relative mean difference (RMD) for each low-level feature. The RMD is a dimensionless measurement of statistical dispersion equal to the absolute mean difference divided by the arithmetic mean. When averaged across the whole set of vocal postures used during training, this measure provides a good estimate of the sensitivity of features to noise. Only features with an average RMD below a threshold are selected, and the rest are rejected because of their significant variation when computed on postures, which, by definition, are perceptually invariant vocal sounds. To measure the amount of nonredundant information embedded in the set of vocal gestures, we measure the intrinsic dimensionality of the matrix containing the selected features computed on all instances of vocal gestures in the training set. We define a quality measure based on a combination of the intrinsic dimensionality and the robustness computed over a large set of computational conditions, and we choose the set of features and parameterizations that maximize the quality measure. We have observed that similar training data from different users determine different features and parameters, as do different training data from the same user. Thus, the optimal set of low-level features is case-specific and must be adaptive both to the user and to the specific control style. Mathematical details are given in earlier work (Fasciani 2012).

**The Vocal Gestural Controller**

Generally, the embedded dimensionality of the training data, which includes the selected low-level features computed on the provided examples of vocal gestures, is significantly lower than the number of features selected from the previous stage. The aim is to represent the vocal data with the smallest number of dimensions and then to linearize the dynamics relating vocal gesture to low-level features. For simplicity, our target space is an N-dimensional \([N-D]\) hypercube (e.g., a square in two dimensions, a cube in three dimensions) that in the next stage is remapped onto the specific timbre space of the synthesizer.

The exclusive use of dimensionality-reduction techniques can produce nonaccessible subregions of the original space, nonlinear response characteristics, and discontinuities between gesture and control signals, all of which may be detrimental to the synthesis control purpose. To address these...
Figure 2. Examples of SOM lattice weights in two dimensions (a) and three dimensions (b), represented as Isomap components. The upper graph in both examples represents the state at initialization, and the lower graph is after training. At initialization, the vertices of the lattice are collocated with the detected gestural extrema, and then pulled towards these points during the training to prevent folding, which could cause sonic discontinuities when mapping gestures during performance.

issues we train an N-D SOM with the gestural training data reduced by Isomap [cf. Tenenbaum, Silva, and Langford 2000]. We then use the SOM as a control structure generating N independent control signals. These are computed by interpolating the lattice indexes of the SOM nodes closest to the vector of low-level instantaneous features of the voice. This approach provides effective control only if the trained SOM preserves the structure of the manifold embedded in the training data without distortions, such as lattice twisting and edge curling. This behavior is not guaranteed by the standard SOM training algorithm. Moreover, the SOM requires that parameters, such as the number of nodes, number of training iterations, and learning and attraction rates, be chosen. These choices can generate distortions if not carefully made. In our training procedure [detailed in Fasciani and Wyse 2013] and briefly illustrated in Figure 2, these parameters are adapted to the size of training data. The modified training algorithm and the prior stage of dimensionality reduction pulls the vertices of the lattice towards the gestural extrema detected in the training data, avoiding folding or twisting, thereby providing a distortion-free lattice adapted to the local distribution in the training vocal data.
In summary, the computation of the vocal gestural controller output from the live vocal input is performed as follows:

1. Compute the low-level features from the voice signal and reject noisy features.
2. Apply the Isomap dimensionality reduction derived from the training data.
3. Compare the resulting $N$-D vector with the SOM lattice nodes.
4. Spatially interpolate the relative indexes of the nodes using the inverse distance weighting (IDW) metric and normalize to unit range.

To provide smoother transitions and discontinuity-free control signals, we limit the SOM search to the $3N$ immediate neighbors of the closest node in the previous iteration. When the input signal is absent or different from those used for training, the controller does not generate any output.

**Computing and Interacting with Synthesis Timbre Spaces**

At this point, the control signals computed from the voice could simply be mapped onto a few synthesis parameters. To provide a richer sonic interaction, however, we propose a strategy to drive a larger number of parameters by mapping control to the synthesis timbre space, from which we derive the synthesizer parameters. Methods to interact with timbre instead of synthesis parameters have been proposed from the pioneering work of Wessel (1979). Such methods embed the perceptual characteristic of the instrument’s timbre in the control-to-parameters mapping. Arfib et al. (2002) formalized this approach in mappings that include intermediate perceptual layers, thereby enhancing the sensitivity and efficiency of the interface, but a generic method to compute and map to synthesizer-specific perceptual spaces has not yet been proposed.

Assuming no prior knowledge of the synthesis algorithm, but assuming a deterministic response, we estimate the model relating parameters to sound by observing inputs and then “listening by machine” to the sonic output. The analysis procedure includes computation of the set of unique vectors of synthesis parameters, which are sent one at a time to the sound generator while computing a set of perceptually related audio descriptors on the sound output. Parameter vectors and associated timbre descriptor vectors are collected in two databases, which together model the synthesis timbral response.

Analyzing all possible parameter combinations with a fine resolution requires excessive time and memory. The model we compute considers only those parameters that users intend to vary, leaving the other parameters fixed. Moreover, we use a coarse parameter resolution in the analysis stage, for which we later compensate by using temporal and spatial interpolation. To provide a comprehensive study of the timbre, for each parameter combination we compute descriptors over multiple overlapping windows of the synthesized signal. These are further analyzed and processed using three different user-selected methods [cf. Fasciani 2016 for details]. In particular, we provide techniques to analyze steady, dynamic, and decaying timbres. The analysis and the following mapping strategy are independent of the specific descriptors used in the system [the current prototype uses the loudness of the 24 Bark critical bands, however].

**Organizing and Mapping the Timbre Space**

After the analysis stage, we have two high-dimensional spaces, relating synthesis-parameter vectors to sound descriptors, and we aim to use spatial coordinates of the timbre space to drive the synthesizer. The dimensionality of the timbre space is high, whereas the output of the vocal gestural controller is at most three-dimensional. Hence, we reduce dimensionality of the timbre space to a number of components $N$ identical to that of the vocal gestural controller. The reduction is performed with Isomap to preserves the geodesic distance in the reduced space, providing greater accuracy in the low-dimensional representation of the timbre than that computed with linear techniques [cf. Burgoyne and McAdams 2007]. Over a large set of experiments with four to eight variable parameters, we
observed that two to four dimensions are generally sufficient to cover a significant degree of the total timbre variance. The top two graphs in Figure 3 give examples of reduced timbre spaces from two different synthesizers subject to the variation of eight synthesis parameters. It is evident that each synthesis case presents a unique shape and distribution of data points. There are regions with either extremely low or extremely high densities. A linear mapping of the gestural controller’s output onto the reduced timbre space would have strong limitations, such as spanning empty subregions of the timbre space that have no computable response. The spatial interpolation in high-density regions increases the chance of driving the synthesizers to inaccurate timbres, owing to the fact that the relationship between parameters and sound is not bijective. To handle such cases, we use a nonlinear mapping that projects the output from the gestural controller onto the timbre space.

For this timbre-space mapping we take a different approach than for the vocal gestural controller, because the synthesis algorithm generating the sound is strictly deterministic, and—as opposed to the voice—the mapping will not be required to respond to new data that were absent during training. First, we redistribute the entries in the timbre space to create a uniform distribution using an iterative technique based on the Voronoi tessellation (Nguyen et al. 2009). The tessellation provides a rigid structure that we progressively deform but do not modify, preserving the neighborhood relations of the data points [i.e., a homomorphic transformation]. The inverse of this redistribution process is the transformation that maps a generic control space \((N-D)\) hypercube to the specific synthesis timbre space, maximizing their relative overlap. Therefore, we use a feed-forward ANN to learn the inverse of the redistribution that represents the function mapping the gestural controller’s output to timbre space. The size of the single-layer ANN is gradually increased until the mapping is sufficiently accurate, measured by combining the mean squared error of the nonlinear regression and the percentages of inaccessible timbre points in a simulated use. In the two bottommost graphs in Figure 3 we show a mapping of the uniform control space through the trained ANN, showing similar shape and distribution with the original timbre space.

The ANN maps the output of the gestural controller onto the reduced timbre space, and the synthesizer is driven with the parameter vector associated with the closest timbre data point. To cope with the limited parameter resolution used in the analysis stage, we apply IDW interpolation. To further smooth the synthesis control, we also use linear interpolation between two consecutive parameter vectors. This synthesis control method has an intrinsic drawback in the case where multiple sets of synthesizer parameters generate the similar sounds. Mapping similar sounds to widely varying parameter values can cause audible glitches during synthesis. As discussed before, this issue is also aggravated by spatial and temporal interpolation. To address this problem we restrict transitions in the timbre space to those points that minimize parameter discontinuities, dynamically locating those points that map to neighboring parameter vectors. We also allow users to define the maximum instantaneous leap in the parameter space, so that they are in control of the trade-off between parameter continuity and the “browsability” of the timbre space.

Functional Prototype

The entire system is implemented as an open-source prototype, which has been optimized to run in real time on general-purpose personal computers and has been used for live performances. The prototype and demonstrations are available online [http://stefanofasciani.com/vci4dmi.html]. The prototype is implemented using Max with a backend engine running in MATLAB. For the runtime interface, a Max patch with a GUI provides users with options to customize the synthesis analysis, the voice preprocessing, and the runtime mapping. A separate Max patch implements the analysis of the synthesizer, and the mapping computation is performed by two offline MATLAB scripts, one for the vocal gestural controller and another for the synthesis timbre space. These generate two data structures including the mapping for the vocal
Figure 3. Examples of timbre spaces in 2-D (a) and 3-D (b) reduced by Isomap. For both types of representation, the topmost graph displays synthesizers subject to the variation of multiple parameters. The middle graphs have entries uniformly redistributed, and the bottommost graphs show the trained artificial neural network (ANN) mapping to the timbre space.
control and for the sound synthesizer, which can be loaded independently to the Max patch at runtime. This separation of functionality allows the same vocal gestural controller to be used with different synthesizers and vice versa.

The procedure for setting up the system does not require technical expertise and the mapping computation is completely automated. We also provide utilities to establish communication between the system and any synthesizer hosted in digital audio workstations such as VST plug-ins using the Open Sound Control (OSC) protocol. Users are required to identify the specific synthesizer parameters they intend to control, with their respective ranges and analysis resolutions. The vocal training data is provided with audio files. There is no minimum requirement on the size of the training data, but with larger sets a more accurate model can be trained. A usable vocal gestural controller can typically be produced with as little as one minute of training data, which includes an equal share of vocal postures and vocal gestures. Because the system is trained using machine listening and unsupervised learning techniques, the user is freed from the burden of providing additional data relating explicitly to the mapping. This implies that users are required to familiarize themselves with the generated mapping, and we provide visual feedback of vocal and timbre spaces to support this task. Moreover, we provide options to further tune the voice-to-synthesis control at run time without the need to repeat the training. We allow users to invert the orientation of the axes of the intermediate space, mapping the output of the vocal gestural controller to the low-dimensional synthesis timbre space. In the mapping computation, there is no explicit criterion we use to set the reciprocal axis orientation. This choice determines a significant change in the response, such as changing the association between voice and synthesis timbre, without adding computational overhead. Other options include the IDW interpolation coefficient, the maximum instantaneous leap in the vocal and synthesis parameter spaces, and the possibility to directly map the output of the gestural controller to the uniformly redistributed timbre space or to the original timbre space, bypassing the ANN mapping. These options enable users to gradually tune the system between two radically different types of vocal interactions: [1] an explicit voice-to-synthesis timbre remapping, or [2] a mapping of voice variations to directions and depth to browse the perceptual timbre space of the sound synthesizer. In Figure 4 there are two examples of the first vocal interaction type, which seems more intuitive, owing to the high correlation between variations of voice and synthesis timbre. Other optional utilities we provide to enhance the interaction with the sound synthesizer include system activation based on voice energy or on similarity to training voice, pitch-tracking-based note generation, and settling to a user-defined state when inactive (i.e., no input voice detected). These user-defined customizations extend the capabilities of the system, are all intuitive and done through listening, and require neither understanding of the mathematics of the mappings nor retraining computation.

The computational performance was evaluated on a 2.4-GHz quad-core Intel i7 processor, with a 1.3-GHz bus speed and 16 GB of memory, running on Mac OS X. Training time depends on the amount of vocal data and on the synthesis analysis settings provided by users, because the size of the SOM and ANN scale with the amount of training data. A typical case that provides sufficiently accurate and stable control uses one minute of vocal training data and about 2,000 synthesis parameter combinations. The resulting analysis and training time is about 45 min for a 2-D gestural controller and timbre space and about twice that for the 3-D case.

The interface latency has two components: the time to acquire one segment of a voice signal and the mapping computation time. The first component depends on the length of the voice segment we analyze to compute feature vectors, which in turn depends on the sampling rate and window size that are automatically selected as their case-specific optimal values. Length values range between 10 msec and 42 msec, with 21 msec being the value most commonly selected by our algorithm. The computation time, from the generation of low-level features to the transmission of the OSC message including synthesis parameters, depends on the system complexity, determined by the amount of data provided for training. On typical cases, the
Figure 4. Two examples of vocal control of sound synthesis, each with the synthesizer sound spectrogram on top and the driving vocal gesture spectrogram below.
computational latency is approximately 1 msec for the 2-D case and 2 msec for the 3-D case. This enables a large window overlap and a throughput of up to 1,000 synthesis parameter updates per second. We measured a latency of 10 msec in a worst-case 3-D scenario tested with 1,728 nodes in the gestural controller, 18 and 6 neurons in the two layers of the ANN, a neighborhood graph with 3.5 million entries for the nonlinear dimensionality reduction, and 15,000 data points in the timbre space. Total memory requirements have never exceeded 3 GB.

**User Study and Evaluation**

It is not possible to define a generic strategy for the evaluation of sonic controllers or musical interfaces (cf. Fels 2004). The methods used in this field involve conflicts and discrepancies, and some are of limited scope (cf. surveys in Marquez-Borbon et al. 2011; El-Shimy and Cooperstock 2016). The evaluations are often informal, idiosyncratic, or not performed at all (Barbosa et al. 2015). Traditional methods used to evaluate human–computer interaction are only effective to compare similar systems or to assess performances on specific tasks (Orio, Schnell, and Wanderley 2001). Limiting the evaluation to a set of predefined tasks may restrict the view on original and creative use, obscuring the different concepts of controllability and sonic expressivity (Dobrian and Koppelman 2006). Perception of benefits and limitations may depend on the aims of individual evaluators. In evaluations centered on the creative and performative use of an interactive system, the recruitment of subjects can be a difficulty if the available population is small, especially when the design addresses users with specific backgrounds and expertise (Wanderley and Orio 2002).

Evaluation strategies for small sample sizes should focus on the identification of trends and patterns, because both statistical measurement and generalizations can be misleading or inconsistent. Moreover, short evaluation sessions are sufficient to assess only basic usability factors, whereas estimating creative and exploratory affordances requires observations over longer periods (Gelineck and Serafin 2009). The design of mapping techniques is the rational outcome of an engineering process, often driven by creative design. The final results can be subjective and criticized, but a scientific methodology to verify the initial engineering principles is still important (Wanderley and Depalle 2004). The user-centered evaluations are essential to determine how a novel interface is received, whether it enables creativity and expressivity, and how it imposes or suggests new modes of thinking about, interacting with, and organizing time or texture in music. The adaptability of a generative mapping and its expressivity are hard to measure objectively. These depend strongly on users’ intentions, which may vary significantly across subjects. Expressivity does not directly depend on the number of controllable parameters, however, but on the range of available choices (Clarke 1988) and on the dimensionality of the control space (Pressing 1990). Furthermore, subjective evaluation can also be strongly influenced by the specific sound synthesis selections when attempting to evaluate interactive mapping only (Gelineck and Serafin 2009). Formal user-centered evaluations do not guarantee effective interaction design, though they can certainly identify poor ones or eventual drawbacks and limitations.

Based on these considerations, the approach we take for evaluation includes both qualitative and quantitative methods. We sample a small group of participants with specific expertise in the field. We carry out a long evaluation session, which includes free exploration of the system, use cases, and open-ended in-depth interviews. These are analyzed to identify trends supporting or modifying our original claims, to point out limitations and possible improvements to our current system, and to aid in developing general principles for applications beyond our immediate aims.

**Method and Experimental Setup**

The mapping between voice and sound synthesis provided by our system depends on the mapping computed from training data, which is case-specific and generated by users. As noted earlier, open-ended systems and frameworks are challenging to validate because of the lack of obvious evaluative criteria.
Table 1. Profiles of Participants in User Studies

<table>
<thead>
<tr>
<th>User ID</th>
<th>Age (years)</th>
<th>Nationality</th>
<th>Experience</th>
<th>Instruments</th>
<th>Instrument Builder</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21–29</td>
<td>India</td>
<td>6</td>
<td>acoustic percussion</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>21–29</td>
<td>Singapore</td>
<td>4</td>
<td>bass, live electronics</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>21–29</td>
<td>China</td>
<td>12</td>
<td>guitar with effects</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>30–39</td>
<td>Singapore</td>
<td>5</td>
<td>live electronics</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>21–29</td>
<td>Singapore</td>
<td>10</td>
<td>piano, saxophone</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>50–65</td>
<td>Scotland</td>
<td>40</td>
<td>synthesizer keyboard</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>30–39</td>
<td>Malaysia</td>
<td>4</td>
<td>live electronics</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>40–49</td>
<td>Switzerland</td>
<td>8</td>
<td>live electronics, VJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>40–49</td>
<td>Canada</td>
<td>25</td>
<td>guitar, live electronics</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>30–39</td>
<td>Spain</td>
<td>18</td>
<td>live electronics, DJ</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Profiles of participants recruited for our user studies. All participants were male.

[Marquez-Borbon et al. 2011]. Results depend on how participants interpret prototypes, which can be different than expected (Gaver et al. 2009). Our evaluation is not limited to measurements and observation of use cases, but also includes free exploration of the system, from which we attempt to identify user experience, embodied interaction, and value-sensitive design (Harrison, Tatar, and Sengers 2007) by analyzing the audiovisual recordings of the sessions and activity logs. A comprehensive objective evaluation is not possible, but the design and usability principles can be verified in user studies.

The qualitative evaluation is based on the analysis of individual open-ended in-depth interviews with participants we invited to use our voice-to-synthesis control system. We interview each subject before and after using the system, to determine individual perspectives and to identify patterns across subjects. In both the pre- and postusage interviews we use the same discovery-oriented strategy (Kvale 1996; Boyce and Neale 2006), asking open-ended questions that allow the interviewer to probe participant perspectives and experiences with the novel interface and to encourage the respondents to freely answer questions using their own words. The semistructured and conversational format of the interviews allows for unexpected digressions to divert the planned question sequence and follow the participant's interest or knowledge. For the quantitative evaluation, we analyze system signals and activity logs over a set of use cases performed by the participants.

To evaluate the system, we recruited musicians and performers who were familiar with sound synthesis and were experienced with interacting with these devices in creative or performing contexts. We chose an initial sample size of ten subjects, considered adequate for phenomenological research of this kind (Thomas and Pollio 2004), and had an additional six available if needed for achieving thematic redundancy in interview results (Guest, Bunce, and Johnson 2006). Participants were purposely selected to have rich knowledge about the study questions (Gubrium 2012), and they came from a musically diverse range of stylistic backgrounds to provide multiple perspectives while being representative of the general population of potential users (Highhouse and Gillespie 2009). In user studies where respondents were not sampled with a uniform distribution, each participant may use a different “frame of reference” and “standard of comparison” when interpreting questions and making judgments. Therefore, scaling responses to questions using Likert scales may result in striking contradictions when compared with open-ended answers (Ogden and Lo 2012). The relatively low sample size favors the ability to probe participants’ deeper and protracted use of the system, and it increases the pertinence of interview data with the interface experience. A summary of participant profiles is provided in Table 1, where “live electronics”
in the Instruments category indicates the broad category of electronic and digital musical instruments controlled by touch-based interfaces other than the piano-based keyboard.

The user studies were conducted in a sound-isolated room with continuous audio and video recording. To facilitate the exploration, the interaction options and tuning settings were mapped on to a labeled hardware controller with LED feedback, as shown in Figure 5. A large screen was used to display the interactive visual representation of the control and timbre spaces. We used a head-worn microphone with a hypercardioid polar pattern for vocal control that minimized the amount of signal the microphone would pick up from synthesizer sounds diffused from the loudspeakers, as well as any other sound sources. This configuration was the same as that used in live performances, and it allowed users to perform other control tasks with their hands. We included a keyboard and a controller in the setup for playing the synthesizers as in a typical performance environment.

The automatic processes of analyzing synthesis sounds and computing the timbre-space mapping are too lengthy to be included in evaluation sessions. Instead of allowing participants to choose and train the system for a single sound synthesizer, we provided a set of eight synthesizers trained in advance, including 2-D and 3-D timbre spaces that the participants selected just by pressing a button on the interface, shown in Figure 5. This provided all participants with an identical experimental condition and an identical set of choices, which they freely explored and customized based on their personal preferences. The set of synthesizers covered a broad range of sonic timbres and provided various control parameters, including FM synthesis, wavetable-based synthesis, granular synthesis, virtual analog modular synthesis, and physical modeling. Moreover, we also included three cases in
which the parameters were controlling, respectively, a low-pass filter, a guitar-amplifier model, or a delay with reverberator applied to the output of a synthesizer with fixed parameters. The number of variable parameters ranged from two to six, and the data points in the timbre spaces ranged from 388 to 12,664.

Protocol and Interview Guidelines

The first interview gathered information about the profile of the participants including their relevant experience with sound synthesizers and their interaction practices in live performance. We also asked questions about the limitations of their current performance setup, how they proposed to overcome such existing limitations, their previous experience with voice-driven systems, and how they imagined a useful vocal interaction with musical instruments. This was followed by an explanation of the principles of voice-to-synthesis timbre interaction and by a demonstration of the full system. Moreover, we demonstrated the individual system components by visualizing the output of the vocal-gesture controller in a 2-D or a 3-D space, and the audio and visual interaction with the timbre space through a touchpad interface.

In the next step, we collected voice samples and used them to train the personalized vocal-gesture controller. A critical issue for the appropriate training and interaction is the understanding of the working principles of the vocal gesture and posture interface. We explained that postures are continuant vocal articulations, whereas gestures may include any articulation, including simple gliding between postures. From each user we collected eight postures with a length of 5 sec each and several instances of vocal gestures with a total duration of 30 sec. For gesture production we guided the users to span the whole space of vocal articulations they intended to use for control.

Participants were initially given training to gain familiarity with the system and the options provided to tune the two components of the mapping. Then we let them freely use and explore the system and the set of pretrained synthesizers. We engaged with participants only if requested or if significant problems were evident, such as an unresponsive mapping because of a poor choice of settings. Then we asked participants to select their preferred synthesizer and mapping options to perform the three following use case tasks: [1] use four different vocal postures targeting four unique synthesis timbres, each with a duration of about 5 sec, and repeat these in six nonconsecutive attempts; [2] perform vocal gestures producing identical dynamic transformation of the synthesis timbre and repeat these in six consecutive attempts for two different cases; and [3] generate as many different synthesis timbres as possible by vocal interaction within 60 sec.

A follow-up interview concluded the evaluation session. This second interview, which represents the core of the qualitative evaluation, aimed to highlight understanding, experience, and perceptions of advantages and drawbacks related to personalized voice-to-synthesis mapping. In particular, we encouraged participants to first describe and then criticize the system in the light of their expertise with other sonic interaction devices. We asked about possible improvements, modifications, and applications relevant to their own performance contexts. Participants were also encouraged to discuss their impression of the control abstraction (timbre versus parameters), their progress in mastering control skills during the practice session, and the benefit of the visual feedback. We limited the total duration of each session to a maximum of 90 minutes, dedicating approximately 15 to 20 minutes for an interview before the trial and a similar time for the second interview afterwards, and at least 30 minutes for the free exploration of the system.

Interview Analysis and Qualitative Evaluation

The qualitative evaluation of this study is based on the analysis of the recorded participant interviews. We started with a time-stamped transcription of the audio and video recordings. Then the transcription and video were further annotated, normalized into noncolloquial text, labeled by relevant topics, and organized at higher hierarchical levels to

Fasciani and Wyse
synthesize participant answers to the key questions (cf. Kvale and Brinkmann 2008; Bryman 2012), while maintaining time alignment between analysis and raw data. For the analysis we worked in two directions: We looked for answers to specific questions on interaction and usability, but the open-ended questions stimulated conversations in which participants freely expressed their thoughts across different topics. To validate and then generalize the respondents’ information, we used triangulation (Rothbauer 2008), where specific opinions and evaluations of the system were verified by multiple participants with substantially different profiles.

Although there is a clear diversity in the answers, we can identify recurrent trends from the first batch of interviews focusing on prior experience. All participants were familiar with control synthesizers from interfaces where individual parameters were mapped to interface elements, such as push buttons, faders, dials, encoders, or equivalent user interface items implemented on touch screens. Six of the participants used preprogrammed automation to transform synthesis timbres in live performances, not only because of the precision and accuracy in tuning parameters but also because manual execution is often impractical owing to the high dimensionality of the parameter space and the lack of spare bandwidth for that interaction. With only one exception, participants had not regularly used voice-driven systems to address their needs, although seven had tried at least hands-free verbal control, such as speech recognition. The main concern expressed was the reliability of the system in noisy environments. Finally, we noticed that participants’ expertise in music technology correlated strongly with their willingness to use the voice for interaction with digital musical instruments and with the sophistication of their ideas to overcome existing limitations in synthesis control. The less-experienced participants suggested speech recognition for driving instruments with voice commands (e.g., “set parameters x to y”). The others suggested using nonverbal characteristics of the voice, tracking and mapping low-level features onto continuous-valued synthesis parameters. High latency and loudspeaker feedback were among the main concerns, however.

From the second batch of interviews we observed that all participants had a clear understanding of the mapping principles and purposes, with eight subjects also highlighting the key concept of training for adaptation. Beneficial aspects mentioned included high expressivity, intuitive and natural interaction, the extra control layer, wide dynamic timbre range of the synthetic sound, smooth response, and generalizability across specific applications or contexts. Respondents considered the use of such a system as an extension to traditional musical interfaces rather than an alternative, and eight participants considered it beneficial for their specific performance context.

When discussing drawbacks and limitations, we observed diversified answers, sometime contrasting, but all suggesting usability improvements. Some problems were due to users selecting inconsistent system options, which could have been overcome by using default settings or selecting an appropriate tuning. This suggests, as three respondents explicitly stated, that the options exposed to tune the system require simplification, especially for novice users. Fewer settings with more meaningful identifiers are necessary. Four participants expressed the need for cross-user and ready-to-use simple mappings, because controlling only a couple of extra parameters by voice already constituted an improvement over existing practices. Finally, three participants argued that adding an adaptive mapping modality, directly associating voice and synthetic timbre by their similarities, would further simplify usability and learnability.

All participants described the interactive graphical representation of the vocal, control, and synthesis timbre spaces as essential for proper learning of the customized mapping. Nine argued that once familiar with the interaction, the visual feedback was not necessary for performing, or that it could be even distracting, unless presented in a minimalistic form. Nine respondents found the system easy and intuitive at first use, but seven of them were concerned about the possibility of achieving the same level of detailed control they obtain with traditional controllers. All users experienced improvements after a half hour of free practicing, however, and agreed that mastering skills and becoming familiar with
the different interaction modalities supported by the system are possible and would improve control accuracy and intimacy. Interestingly, eight subjects considered the simultaneous multiparametric control more important than absolute control precision for performance expressivity.

One of the most significant emergent themes for vocal control of synthesizer timbre, in general, concerned the user experience. Seven subjects expressed the feeling of interacting directly with a sonic object, so that the experience was one of driving a synthetic timbre via voice timbre without thinking about the synthesis algorithm involved in the process. The remaining three subjects tried to explicitly control the synthesis parameters (rather than the resulting timbre), which resulted in challenges due to the weak bijective parameter-to-timbre relationship. They argued, however, that training and using the system with the synthesizers they were familiar with would allow them to focus on timbral results directly rather than on parameters.

Examining the recordings of the participants freely exploring the vocal control system, we identified several recurring patterns. Nine participants started exploring the responsiveness of the system and then passed to a more systematic phase, in which they used the visual feedback to learn the generated mappings. In particular, they used the control space first and then the timbre space visualizations to understand the interaction provided by each component. Four users experienced difficulties in understanding the tuning options, and they did not explore all available choices. The two options most frequently used in tuning the generative mapping were the maximum instantaneous leap in the parameter space, also called interface “sensitivity” by participants, and the inversion of axis orientation in the intermediate space, mapping the output of the vocal gestural controller to the low-dimensional synthesis timbre space. We did not observe any correlation between the number of synthesis parameters simultaneously controlled by voice and ease or complexity of use. This suggests that the parameter space is completely transparent and hidden from users. Finally, we noticed that when the output sound of the synthesizer had a clear pitch, participants often tuned their voice to it. In some cases, this resulted in the use of vocal signals significantly different from the ones they generated for the training set, which in turn impaired the equal access to subregions of the control space. A design supporting multiple vocal gestural controllers, each associated with a different pitch range, could address this tendency to match voice and synthesizer pitch, as would including a preprocessing stage that tracks and converts the runtime vocal pitch to match training data.

Quantitative Evaluation

The quantitative evaluation is based on the analysis of the configurations participants chose for vocal control of sound synthesis and on the performance metric computed over the three use cases described in the protocol. In Table 2, we summarize the preferred synthesizer and the key mapping options that participants selected to perform the use case tasks.

All participants selected a 2-D gestural controller and timbre space, which is easier for first-time users and provides a more effective dimensionality reduction of the synthesis control space. This also suggests that the increase in timbral-control detail is not worth the slightly higher cognitive complexity required to interact with the 3-D mapping. Eight subjects selected the ANN for mapping from the gestural controller to the timbre space, and the remaining two selected the alternative mapping onto the uniformly redistributed timbre space, which is slightly less accurate in terms of the linearity of timbre variation, but which is also less computationally complex and therefore has a lower response latency. As expected, no participant chose to map the gestural controller to the original timbre space, just scaled to match the ranges, because of the poor usability resulting. This is significant because it suggests that the usability of the system draws a great benefit from the generative mapping implementing the homomorphic transformations across heterogeneous voice and timbre spaces.

The values limiting the instantaneous leap of synthesis parameter space were different and

Fasciani and Wyse

53
Table 2. Synthesizer and Key-Mapping Options

<table>
<thead>
<tr>
<th>User ID</th>
<th>Mapping Parameter Mode</th>
<th>Mapping Parameter Space Leap</th>
<th>Synthesizer Type</th>
<th>Timbre Synthesis Analysis</th>
<th>Timbre Synthesis Parameters</th>
<th>Timebre Synthesis Space Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>URTS</td>
<td>minimum</td>
<td>FM</td>
<td>steady</td>
<td>6</td>
<td>4,800</td>
</tr>
<tr>
<td>2</td>
<td>ANN</td>
<td>0.1</td>
<td>virtual analog</td>
<td>steady</td>
<td>4</td>
<td>1,300</td>
</tr>
<tr>
<td>3</td>
<td>ANN</td>
<td>0.2</td>
<td>guitar amplifier</td>
<td>steady</td>
<td>5</td>
<td>7,200</td>
</tr>
<tr>
<td>4</td>
<td>ANN</td>
<td>minimum</td>
<td>low pass</td>
<td>steady</td>
<td>2</td>
<td>478</td>
</tr>
<tr>
<td>5</td>
<td>URTS</td>
<td>minimum</td>
<td>wavetable</td>
<td>decaying</td>
<td>4</td>
<td>5,300</td>
</tr>
<tr>
<td>6</td>
<td>ANN</td>
<td>0.5</td>
<td>virtual analog</td>
<td>steady</td>
<td>4</td>
<td>1,300</td>
</tr>
<tr>
<td>7</td>
<td>ANN</td>
<td>0.7</td>
<td>virtual analog</td>
<td>steady</td>
<td>4</td>
<td>1,300</td>
</tr>
<tr>
<td>8</td>
<td>ANN</td>
<td>minimum</td>
<td>virtual analog</td>
<td>steady</td>
<td>4</td>
<td>1,300</td>
</tr>
<tr>
<td>9</td>
<td>ANN</td>
<td>minimum</td>
<td>delay and reverberation</td>
<td>dynamic</td>
<td>3</td>
<td>388</td>
</tr>
<tr>
<td>10</td>
<td>ANN</td>
<td>0.4</td>
<td>wavetable</td>
<td>decaying</td>
<td>4</td>
<td>11,600</td>
</tr>
</tbody>
</table>

Options selected by participants for the use cases. URTS = uniformly redistributed timbre space; ANN = artificial neural network. All users opted for 2-D gestural controller and timbre space.

spanned from the minimum to 0.7. The minimum, which is case-specific and depends on the parameter resolution used in the analysis stage, only allows navigation to timbre space points that are also immediate neighbors in the synthesis parameter space, resulting in lower sensitivity but higher control accuracy.

The timbre spaces that included approximately 1,000 to 5,000 data points from the synthesis/analysis stage were preferred and more usable than those outside this range. Spaces with a higher number of points were selected by one participant only. As explained earlier, these may determine a more challenging interaction, especially when a small value of the maximum parameter leap is set, without providing any significant improvement in the timbre accuracy. The participant who selected a timbre space with 11,600 data points also set a relatively relaxed limit on the instantaneous parameter leap, however. This is the selection we would expect, because it determines a better interface response. Participants 4 and 9 selected a timbre space with fewer points. The use of IDW interpolation hid the coarse resolution of the timbre space, however, and this was unnoticed by participants. Seven participants selected synthesizers that generated a steady timbre, for which the specific analysis mode provides the tightest coupling between vocal articulation and synthesis sound, implementing an instantaneous remapping across the sonic domains. Finally, the synthesizer presenting the widest timbre range was the most popular, being selected by participants 2, 6, 7, and 8.

The quantitative evaluation computed over the three use-case tasks attempts to estimate the reliability, usability, and repeatability of the vocal control of sound synthesis. Results are illustrated in the four charts in Figure 6. Participants had no feedback on accuracy, precision, or completion while executing the task. In the first use case, we evaluate the stability of control over vocal postures, measuring the standard deviation of the gestural controller's output mapped onto the synthesis timbre space, as well as the standard deviation of the resulting synthesis parameter. Results showed small values for standard deviation, also when averaged over the multiple dimensions of the gestural controller's output and the synthesis parameter vector. Low-level noisy features were already rejected from entering the system, and the SOM-based approach contributed to improving the control stability. The values for standard deviation could increase slightly in the parameter space, depending on specific tuning options. In most cases the values were still below 0.1, however, meaning that we successfully rejected the intrinsic noises in voice signals. Therefore, vocal postures effectively determined invariant synthesis parameters.
We evaluated the repeatability of static control by measuring the difference between vocal postures that were aimed at generating an identical synthesis timbre. The difference is estimated by comparing the Euclidean distance in the timbre space, which in the worst case is equal to the square root of two, to the Euclidean distance in the synthesis parameter space, which in the worst case is equal to the square root of the number of parameters. Results showed large values for some subjects, especially for the distance between the parameter vectors. We verified that for participant 3 the large distances were due to poor skills in repeating similar vocal postures. For participants 5 and 8, the distance was due to strict limits on the instantaneous parameter leap. Indeed, the average distances in the control space were generally lower, which suggests that the static control of timbre is more repeatable and accurate than the detailed control of parameter values, thereby providing justification.
for mapping control to perceptual timbre rather than to synthesis parameters. Moreover, because different sets of parameters can generate similar or identical timbres, larger distances in the parameter space do not necessarily imply timbre inaccuracy.

In the second use case, we evaluated the repeatability of dynamic control, measuring the differences across instances of identical vocal gestures that were aimed at identical timbre variations. Dynamic time warping was used to equalize the length of the sequences generated by participants across instances. As with postural control, the differences in the timbre space, directly controlled by the vocal gestural controller, were lower than those in the synthesis parameter space. When analyzing starting coordinates, ending coordinates, and overall shape of the timbre control path, eight participants managed to repeat similar patterns across different trials. The trajectories were not identical but showed that it is possible to repeat control-space trajectories. As observed and explicitly stated by participants, accuracy of dynamic and static control can be mastered by practicing with the same configuration for longer periods.

Finally, for the third use case we evaluated the ability to access all nodes of the SOM, which mapped onto different subregions of the timbre space. We also measured the percentage of unique parameter combinations obtained, which were computed by IDW interpolation. Results showed that all participants were able to cover at least 70 percent of the control space. The percentages of unique parameter combinations tended to be lower, and they correlated with the total number of data points and with the maximum parameter leap set by the user. The coverage of the spaces that we detected across all the free-practice sessions was close to 100 percent for all participants. These measurements were not derived from a systematic and structured test, however. The diversity of the results across users is evident, and for participants 3, 4, and 8 this can be partially ascribed to insufficient familiarization with the system, as we observed from the session recordings. For this study we intentionally did not recruit any performers with specific prior vocal training, such as singers, to demonstrate that the system can be used by those with average, untrained skills in vocal articulation.

Conclusion

Using the voice for controlling digital musical instruments is an emerging trend that has interesting practical and creative applications. It also presents many challenges owing to the high dimensionality of timbral spaces and the demands for continuity, expressivity, and customizability that we find in musical contexts. We developed a generic framework for the unsupervised training of a personalized mapping between voice and synthetic sound generation, adapted to the sonic characteristics of the voice and of the synthesized sound. Unsupervised machine listening and machine learning were integrated with each other to fully automate the generation of mappings, while still providing instrument designers and users with the capability to further personalize the interaction. The adaptive mapping strategy maximizes the explorable sonic space of a synthesizer given a timbral domain for vocal control. The resulting interaction minimizes the cognitive complexity for controlling high-dimensional synthesis parameter spaces and, at the same time, minimizes possible timbre losses due to potentially low-dimensional control spaces. The nonlinearity and the poor perceptual relevance of traditional synthesis controllers were addressed by coupling control data with sonic features rather than with the synthesizer parameters themselves.

We evaluated various aspects of the vocal-to-synthesizer timbral mapping system, as well as the overall effectiveness of vocal control of synthesis, with experienced performing musicians in a series of qualitative and quantitative user studies. The system was able to provide reliable and repeatable sonic control that users were able to master after a short period of initial exploration and practice. The musicians achieved a satisfactory level of musical accuracy using only their voice for control. Participants responded favorably to the possibilities and benefits provided by such a control layer in performance contexts when other control channels (i.e., their hands) are fully occupied. The mapping
techniques we developed would also have utility in other high-dimensional interaction contexts beyond the vocal control of synthesizers.

References


Fasciani and Wyse