Motion-Sound Mapping through Interaction

Jules Françoise, Frederic Bevilacqua

Motion-Sound Mapping through Interaction

An Approach to User-Centered Design of Auditory Feedback using Machine Learning

JULES FRANÇOISE, LIMSI, CNRS, Université Paris-Saclay, France
FRÉDÉRIC BEVILACQUA, UMR STMS, Ircam, CNRS, Sorbonne Université, France

Technologies for sensing movement are expanding towards everyday use in virtual reality, gaming, and artistic practices. In this context, there is a need for methodologies to help designers and users create meaningful movement experiences. This paper discusses a user-centered approach for the design of interactive auditory feedback using interactive machine learning. We discuss Mapping through Interaction, a method for crafting sonic interactions from corporeal demonstrations of embodied associations between motion and sound. It uses an interactive machine learning approach to build the mapping from user demonstrations, emphasizing an iterative design process that integrates acted and interactive experiences of the relationships between movement and sound. We examine Gaussian Mixture Regression and Hidden Markov Regression for continuous movement recognition and real-time sound parameter generation. We illustrate and evaluate this approach through an application in which novice users can create interactive sound feedback based on coproduced gestures and vocalizations. Results indicate that Gaussian Mixture Regression and Hidden Markov Regression can efficiently learn complex motion-sound mappings from few examples.

CCS Concepts: • **Human-centered computing → Auditory feedback**; *Interaction design theory, concepts and paradigms*; • **Computing methodologies → Learning from demonstrations**;

Additional Key Words and Phrases: Interactive Machine Learning, Programming-by-Demonstration, User-Centered Design, Sound and Music Computing, Sonification, Movement.

ACM Reference Format:
Jules Françoise and Frédéric Bevilacqua. 2018. Motion-Sound Mapping through Interaction: An Approach to User-Centered Design of Auditory Feedback using Machine Learning. *ACM Trans. Interact. Intell. Syst.* 0, 0, Article 0 (January 2018), 30 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Movement interaction is emerging at the forefront of Multimedia and Human-Computer Interaction (HCI) research. Technologies and contexts of use for motion sensing are constantly expanding, reaching out beyond academic circles towards a general audience. Beyond task-oriented gestural interaction, movement can support human expression and learning in multimedia applications such as virtual reality, serious games, or creative applications in dance and music. Such applications necessitate the development of specific methodologies and frameworks to help designers and users create meaningful movement experiences.

Authors' addresses: Jules Françoise, LIMSI, CNRS, Université Paris-Saclay, Bât 508, rue John von Neumann, Campus Universitaire, F-91405, Orsay, France, jules.francoise@limsi.fr; Frédéric Bevilacqua, UMR STMS, Ircam, CNRS, Sorbonne Université, 1, Place Igor Stravinsky, 75004, Paris, France, bevilacqua@ircam.fr.

In this paper, we focus on designing the interaction between movement and sound, motivated by a large range of applications ranging from performing arts [28] to movement learning or rehabilitation guided by auditory feedback [7]. In these contexts, it is often essential for users to personalize the movement-sound relationship to their context of use and individual needs. In music and performing arts, developing custom gestural languages for interacting with sound and music is necessary to support the artistic discourse. A musician willing to map particular movements to audio processing might design gestures according to the metaphor and poetics of the performance. In movement learning contexts such as rehabilitation, adapting to individuals is critical to fit different abilities and motor skills. As a result, whether sound control is an explicit goal (as in the case of music performance), or sound feedback intrinsically supports the movement execution itself (e.g. for rehabilitation), there is a critical need for users to personalize gesture-sound relationships.

Our general goal is to develop a user-centered framework for designing interactive systems with auditory feedback. In particular, it aims to enable users to design continuous feedback that can be controlled in real-time by their own gesture vocabularies. Users should be able personalize the mapping between motion and sound by demonstrating a set of examples of associated sounds and gestures.

Our approach relies on Interactive Machine learning as a tool for rapid prototyping of the movement-sound relationships. Users can therefore craft interactive systems by iteratively specifying their gestures and their associated sounds, training the system, and directly interacting with the learned mapping, as described by Fiebrink et al. [28]. We emphasize the creation of demonstrations of continuous gestures performed while listening to the corresponding sounds. These multimodal recordings are used to train a probabilistic model of the mapping that can be used for real-time control of sound synthesis based on new movements. In this paper, we propose to use generative models for the real-time generation of sound parameters associated to a movement.

From a conceptual point of view, as we propose a design process based on a short cycle of recording and performing gestures with sound feedback that facilitates the integration of the action-perception loop. This approach therefore supports a shift from designing "for" user experience to designing "through" user experience. We argue that the iterative nature of the design process is essential. Users progressively learn to create effective demonstrations and to execute gestures accurately. As a result, we refer to our approach as "mapping through Interaction", to emphasize not only the role of embodied demonstrations for crafting movement control strategies, but the importance of rapid testing through direct interaction with the trained mapping.

Specifically, we describe in this paper the following contributions

- We propose the use of two probabilistic regression models for multimodal motion-sound sequence mapping, respectively based on Gaussian Mixture Models and Hidden Markov Models. We describe an implementation that allows the user to specify parameters governing the model's complexity and generalization, which can be adjusted to target specific interaction design contexts. Our implementation allows for real-time generation of the sound parameters.
- We describe an application that uses co-produced gestures and vocalizations for the design of auditory feedback. We report the results of a study that compares the performance of the proposed methods of Hidden Markov Regression and Gaussian Mixture Regression with other regression techniques on a dataset of of gestures associated with vocalizations. The study also highlights the learning process of the participants, who iteratively improve the quality of their demonstrations, and the consistency of their movements.

The paper is organized as follows. We review related approaches using interactive machine learning for mapping movement to sound in Section 2. We provide a conceptual motivation of our approach in Section 3, and give a detailed description of the design process of Mapping through interaction in Section 4. Section 5 describes two models for modeling motion and sound relationships: Gaussian Mixture Regression and Hidden Markov Regression. Sections 6 and 7 respectively describe and evaluate a specific application of the approach that uses user-defined gestures and vocalization to create interactive sound feedback. Finally, Section 8 provides a general discussion of the proposed approach.

2 RELATED WORK

In this section, we review related works on interactive machine learning that reconsiders the role of the user in machine learning systems. We detail how interactive machine learning can support the design of movement interaction, in particular regarding motion-sound mapping design within the New Interfaces for Musical Expression community.

2.1 Interactive Machine Learning

Machine learning's ability to learn representations from real-world data represents one of the most promising approaches for designing movement interaction. In classical machine learning workflows, however, user engagement is limited. The training data is fixed and users' only opportunity for intervention is to evaluate the results of the training, or the classification of new input. Yet, using machine learning for interaction design requires more expressive frameworks that provide a deeper involvement of users in the design and evaluation process.

The framework of Interactive Machine Learning (IML), proposed by Fails and Olsen [24], introduced a fluid interaction workflow that integrates users at all steps of the process, from providing training data to training and evaluating machine learning models. Since then, several lines of research have focused on improving user interaction with machine learning systems, using for example user feedback on target concepts in recommender systems [29], better programming environments [28, 69], visualization [1, 80], or pedagogical approaches that explain the system's decisions to users [58].

Such user-centered approaches to machine learning emphasize a fluid workflow that can allow novice users to design efficient classifiers or regression models for interaction based on movement and gestures [91]. For example, the Wekinator [28] encourages iterative design and multiple alternatives through an interaction loop articulating configuration of the learning problem (selection of features and algorithm), creation/editing of the training examples, training, and evaluation. Through several studies, Fiebrink et al. showed that users consistently iterate over designs, analyzing errors and refining the training data and algorithms at each step [28]. Similarly, IML can facilitate the integration of user-defined gestures for communication with virtual characters [42], or to reinforce bodily interaction in gaming [56].

2.2 Interactive Machine Learning for Motion-Sound Mapping

Designing the relationship between movement and sound has been central to research and practice of interactive audio and music applications, in the fields of New Interfaces for Musical Expression (NIME), Digital Musical Instrument (DMI) design [63], Sonic Interaction Design (SID) [37], and sonification [47]. The design of the *mapping* [74] between motion and sound highly determine the interaction possibilities, for example in terms of ease-of-use, expressivity, controllability, or metaphorical qualities. Mapping in the NIME community has evolved from *explicit* wiring of parameters toward *implicit* mapping strategies that use an intermediate interaction model [50]. Such models can take a variety of forms, such as interpolation maps [46, 87, 88] or dynamical systems [45, 52, 65]. Interactive Machine learning (IML) is particularly relevant to implicit mapping design in that it allows instrument designers and performers to express personal and individualized movements. Its application to motion-sound mapping design has focused on two major approaches, respectively based on gesture recognition and regression [19].

With gesture recognition, one can link the identification of particular gestures, which might carry a semantic meaning, to the control of audio processing. Many machine learning algorithms have been applied to real-time gesture recognition, for example Hidden Markov Models (HMMs) [6, 57], Dynamic Time Warping (DTW) [6], and Artificial Neural Networks (ANNs) [12]. While many methods for real-time gesture recognition have been proposed and distributed in the NIME community [40], most uses of gesture recognition are confined to discrete interaction paradigms such as triggering a musical event when a particular gesture is recognized [41].

In sound and music computing, complex mappings involving many-to-many associations are often preferred to simple triggering paradigms. To address the need for continuous interaction in music performance, gesture recognition has been extended to characterize gestures as continuous processes varying in timing and dynamics. *Gesture Follower* [9] is built upon a template-based implementation of HMMs to perform a real-time alignment of a live gesture over a reference recording. The temporal mapping paradigm introduced by Bevilacqua et al. exploits this real-time alignment to synchronize the resynthesis of an audio recording to the execution of a gesture [8]. *Gesture Variation Follower* (GVF) [17] extends this approach to the tracking of several features of the movement in real-time: its time progression but also a set of *variations*, for example the offset position, size, and orientation of two-dimensional gestures. Caramiaux et al. showed that the model consistently tracks such gesture variations, which allows users to control continuous actions. However, both approaches are confined to single-example learning, which limits the possibility of capturing the expressive variations that intrinsically occur between several performances of the same gesture.

Alternatively to gesture recognition, supervised learning can be used to directly learn the mapping between motion and sound parameters through regression. Many approaches rely on neural networks, that have a long history in machine learning for their ability to learn the characteristics of non-linear systems [25, 26, 59, 64]. While neural networks are a powerful tool for non-linear mapping design, training such models can be tedious, notably because of the lack of transparency of the training process. Moreover, most implementations for interaction are lacking an explicit temporal model that takes into account the evolution of movement features over time.

In this paper, we extend and generalize an approach to regression based on probabilistic sequence models [34]. We formalize a general framework for cross-modal sequence mapping with Gaussian models, and we propose two models for regression that rely on joint probabilistic representations of movement and sound. The first model is based on Gaussian mixtures and can be seen as a static regression. The second integrates a multilevel sequence model derived from Hidden Markov Models that help creating continuous, time-evolving, relationships between movement and sound parameter sequences.

2.3 Probabilistic Models: from Recognition to Synthesis

In this section, we make a detour through speech processing, animation and robotics. We briefly review approaches to cross-modal mapping that rely on generative sequence models.

2.3.1 Statistical Parametric Speech Synthesis. The rapid development of statistical speech synthesis has been supported by well-established machine learning method from speech recognition. As a matter of fact, Hidden Markov Models (HMMs) provide a unified modeling framework for speech analysis and synthesis, allowing to transfer methods — such as speaker adaptation, — from recognition to generation. Statistical parametric speech synthesis provides a flexible framework for expressive synthesis, that can integrate prosodic, articulatory of affective features [85]. While several methods have been proposed for efficient and robust parameter generation with HMMs [86, 93], most of them are oriented towards offline generation.

Novel applications such as speaker conversion or automated translation are encouraging a shift from pure synthesis towards the issue of sequence mapping. *Acoustic-articulatory mapping* — also known as *speech inversion* — aims at recovering from acoustic speech signals the articulation movements related to vocal production. Toda et al. proposed to use the Gaussian Mixture Models (GMMs) for regression [83], following the theoretical work of Ghahramani and Jordan [39]. The method consists in learning a joint multimodal model from observation vectors built as the concatenation of the input and output feature vectors. For regression, each Gaussian distribution is expressed as a conditional distribution over the input features. Subsequent work applied HMMs to the problem of feature mapping, extending the trajectory HMM to learn a joint model of acoustic and articulatory features [49, 92, 94].

2.3.2 Movement Generation and Robotics. As for speech synthesis, statistical Gaussian models have proved efficient for encoding the dynamics of human motion. Their flexibility allows for generating movement according to external variables such as style. Brand and Hertzmann's 'style machines' extract style parameters implicitly at training, which can then be used during generation for interpolating between styles [11]. Drawing upon speaker adaptation methods in speech processing, Tilmanne et al. proposed a method for style adaptation and interpolation in walking motion synthesis [81, 82].

In robotics, several methods for motor learning by demonstration draw upon probabilistic models for motor task modeling, reproduction and generalization. Calinon et al. proposed to learn a joint time/motion GMM trained from multiple demonstrations of a motor behavior [15]. They further extended the method by combining HMMs with GMM regression (or Gaussian Mixture Regression, GMR) where the weight of each Gaussian are estimated by a forward algorithm [14]. The method was shown to outperform the time-based GMR, and gives similar results as Dynamic Movement primitives [51], but can more easily learn from several demonstrations with variations.

We propose a similar framework for user-centered design of motion sound mapping. Our method draws upon the use of Gaussian models for sequence-to-sequence mapping. Importantly, the context of interaction design required to adapt the model for training on few examples and real-time generation.

3 CONCEPTUAL MOTIVATION

Several threads of research in Human-Computer Interaction (HCI) have addressed movement interaction through different lenses, ranging from task-driven accounts of gestural interaction to experiential approaches relying on somatic practices and expert knowledge. In this section, we motivate the Mapping through Interaction framework by considering the importance of the action perception loop, in particular with regards to recent results in embodied music cognition.

3.1 Movement Design and Embodied Cognition

Embodied cognition theories emphasize the essential role of the body in cognitive phenomena. Embodied cognition supports that knowledge, reasoning and behaviors emerge from dynamic interactions within the environment [2]. The theory of embodied cognition has a significant impact on current trends in HCI research [54] and interaction design [23], by reassessing the role of the body in computer-mediated interaction.

Yet, gesture design remains a challenging task. Participatory design is now a standard approach for gesture creation through elicitation studies with novice users [20, 90]. While such approaches to gestural interaction design are leading the way to end-user adaptation and customization, they can be limited by several factors. User elicitation of gesture sets often suffers from a "legacy bias" that conditions users in creating gestures that mimic their previous experience with technology [90]. Misconceptions about sensing technologies or recognizers' abilities are often limiting, and can be critical for end user gesture customization [67]. Moreover, the limitation to discrete interaction paradigms often overlooks the nuance and expressiveness of movement. Alternative methodologies reconsider how we can design movement interaction through somatic practices [48, 62, 77]. By adopting a more holistic and situated approach to movement design, these techniques can limit the legacy bias and result in more engaging interactions. Such experiential approaches promote the value of designing for and through movement.

Most approaches, however, do not heavily rely on a given feedback loop during the design process. Yet, the action-perception loop is necessary to appreciate the nuances of movement execution, which make it such an efficient and expressive modality in human interaction and communication. Pointing devices such as the mouse are extremely efficient because they rely on quantitative models of movement execution in a restricted design space. Their success is due to a large body of work on the implementation of a tight visuo-motor loop. The action-perception loop enables learning and skill acquisition, and it is therefore essential to consider in movement interaction.

We focus on the design of movement interaction with auditory feedback. In music, Leman suggest that listeners engage with listening through motor simulation, putting bodily experience as a primary vector of musical expression [60]. Experiments investigating spontaneous motors responses to sound stimuli show that people present a wide range of strategies for associating gestures to sounds, such as mimicking the sound-producing actions [16, 44], or tracing the perceived properties of the sound [43]. All studies underline that associations between movement and sound are highly idiosyncratic, in light of the large diversity of gestures generated in response to sound. This suggests that sound feedback could support the creation of gestures, but also that the design tools need to adapt to the variability induced by contextual, cultural, and personal factors.

3.2 Conceptual Approach

Approaches to motion-sound mapping design have evolved from analytical views of the mapping between motion and sound parameters towards approaches based on the action-perception loop at a higher level. At the same time, the recent developments of interactive machine learning support data-driven design approaches that allow users to design interactions by example. A particularly interesting methodology is *play-along* mapping [27], which relies on a *score* of sound presets to guide to the definition of the training examples. The approach, however, might require to define the score manually, and does not explicitly considers listening and perception as a starting point.

Our approach for designing sonic interactions draws upon the general principle of *mapping through listening* [18], that builds upon related work on listening modes and gestural sound descriptions to formalize three categories of mapping strategies: *instantaneous*, *temporal*, and *metaphoric*. *Instantaneous* mapping strategies refer to the translation of magnitudes between instantaneous gesture and sound features or parameters. *Temporal* mapping strategies refer to the translation and adaptation of temporal morphologies (i.e. profiles, timing, and event sequences) between the gesture and sound data streams. *Metaphorical* mapping strategies refer to relationships determined by metaphors or semantic aspects, that do not necessarily rely on morphological congruences between gesture and sound.

Mapping through listening is not a technical approach but a design principle that considers embodied associations between gestures and sounds as the essential component of mapping design. We propose to explicitly consider corporeal demonstrations of such embodied associations as a basis for learning the mapping between motion and sound. We combine the design principle of mapping through listening with interactive machine learning in a framework we call *Mapping through Interaction*.

4 MAPPING THROUGH INTERACTION

We define Mapping through Interaction as a method for crafting sonic interactions from corporeal demonstrations of embodied associations between motion and sound. It uses an interactive machine learning approach to build the mapping from user demonstrations, emphasizing an iterative design process that integrates *acted* and *interactive* experiences of the relationships between movement and sound.

The term Mapping through Interaction evolved from our initial definition of *Mapping-by-Demonstration* [30] that referred to the field of Programming-by-Demonstration in robotics [3].[1] Robot programming-by-demonstration focuses on reproducing and generalizing behaviors from a set of demonstrations from a human teacher. Hence, it emphasizes the role of the human in the demonstration and specification of desirable behaviors. Our goal is to emphasize not only the role of embodied demonstrations for crafting movement control strategies, but the importance of rapid testing through direct interaction with the trained mapping.

4.1 Overview

We now give an overview of the workflow of the Mapping through Interaction framework from a user's perspective, as illustrated in Figure 1. The framework implements an interaction loop iterating over two phases of *Demonstration* and *Performance*.

The demonstration phase starts with a listening phase in order to generate movement. The user imagines a movement that can be associated to a particular sound (**listening**). Then, this embodied association between motion and sound needs to be acted. The user provides the system with an example of the mapping, by performing a movement along the sound (**acting**). We record the motion and sound parameter streams to form a synchronous multimodal sequence, which constitutes a demonstration of the movement-sound mapping. The aggregation of one or several of these multimodal demonstrations constitutes a training set, which is used to train a machine learning model encoding the mapping between motion and sound. Once trained, this mapping model can be used in the *Performance* phase. The user can then reproduce and explore the created mapping through movement (**performance**). Movement parameters are then continuously streamed to the mapping layer that drives the sound synthesis, giving a direct feedback to the user (**listening**). This feedback serves as material to reflect on the design. It allows users to compare the interactive relationship, as learned by the system, with the initial embodied association that was acted in the demonstration. Thus, this framework allows users to quickly iterate through a design process driven by the interaction within a feedback loop.

4.2 Design Process from the User's Perspective

The design process can be broken down to a sequence of actions available to the user: (1) create a set of sounds, (2) record demonstrations of gestures synchronized with sounds, (3) edit and annotate the resulting training data, (4) configure and train the machine learning model, (5) evaluate the learned mapping through direct interaction:

[1] Note that *imitation* learning is also widely used in robot motor learning [75, 76]. Although the term is particularly relevant in humanoid robotics, its application to the problem of motion-sound mapping reduces the scope to having a computer imitating a human, which is not the purpose of the proposed framework.

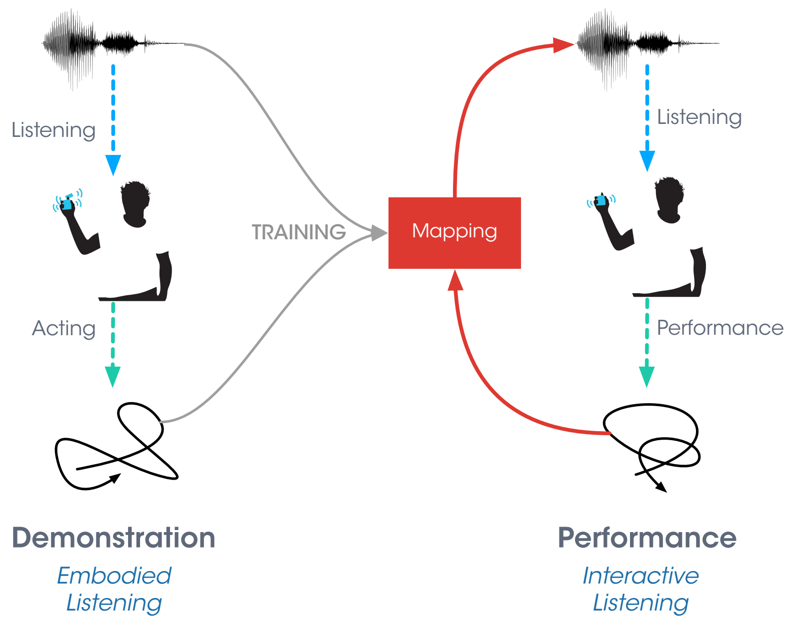


Fig. 1. Overview of the workflow of Mapping through Interaction. Blue and green dashed arrows respectively represent listening and moving. In *Demonstration*, the user's movement performed while listening is used to learn an interaction model. In *performance*, the user's movements continuously control the sound synthesis with the learned mapping.

Sound Design. Users first need to create a set of sounds to be used as demonstrations. These sounds can either be synthesized, or selected from a corpus of recorded sounds. In the case of parametric synthesis, users create sounds by editing trajectories of sound control parameters, as we previously proposed with physical modeling sound synthesis [34]. In this paper, we focus on sampled-based sound synthesis (as detailed in the section). In this case, users can create a selection of recorded sounds from an existing corpus.

Data Creation by Demonstration. All examples used to train the machine learning model are provided by the user, who can record gestures synchronized with sounds. Gestures are captured using motion sensors such as inertial sensors. The choice of appropriate sensors and movement features depends on the application, as discussed in the next section. Our methodology derives from the Mapping through listening approach that considers modes of listening as the starting point for interaction design. Recording a gesture while listening to the associated sound gives users the opportunity to enact the motion-sound relationship, and adjust their gesture if necessary. Moreover, this ensures the correct synchronization of both modalities.

Data Editing and Annotation. Machine learning from few examples is challenging and requires high quality demonstrations. Since the demonstrations are not always perfectly executed, it is useful for to give users the possibility to select or improve the segmentation of some of the recorded demonstrations. In addition, demonstrations can be annotated with a set of labels representing various *classes*. Classes are ensembles of gestures used for joint recognition and regression. For example, one could record several variations or a circular gesture associated to recordings of water sounds, and several variations of a square gesture associated with industrial sounds. In performance, we can use the trained machine learning model to jointly recognize the class (circle or square), and generate the associated sound parameters by regression.

Machine Learning Configuration and Training. Users can interactively train the machine learning on their set of demonstrations. We let users directly manipulate some of the parameters of the machine learning models that deal with regularization or model complexity. These particulars parameters are described and discussed in detail for the proposed probabilistic models in Section 5.3.

Evaluation by direct interaction. Finally, users can directly interact with the learned mapping in the *Performance* phase. Movements performed at the input of the system are recognized and mapped in real-time to the auditory feedback. This allows users to evaluate the quality of their design: they can reproduce their demonstration gestures to assess the consistency of the feedback or the reliability of the gesture recognition, or they can perform new variations of their gestures to evaluate the generalization of the mapping. Such direct interaction is essential to reflect on the mapping and modify any of the previous actions.

Note that this workflow can be adjusted to the target users. We have applied this approach both in public installations involving novice users, as further described in the use-case of Section 6, and in collaboration with expert composers [4]. When presenting systems as interactive installations, sound design, annotation and machine learning configuration was hidden from end users for simplicity. We further discuss the importance of expertise in Section 8.2.

4.3 Technical Overview

In this section, we detail the main technical building blocks necessary to implement a Mapping through Interaction system: movement capture and analysis, sound analysis and synthesis, and machine learning.

Motion Capture and Analysis. Capturing and analyzing users' movements is essential in the implementation of a mapping through interaction system. The choice of sensors and the feature extraction method should be chosen according to the type of gestures that are meant to be captured. In our work, we mostly focus on wearable sensors such as inertial sensors (accelerometers, gyroscopes) that can be handheld or body-worn. We experimented with different features according to the scenario of use, from smoothed acceleration signals to higher-level features. In this paper, we represent the movement with frames of low-level features from accelerometers sampled at 100 Hz.

Sound Analysis and Synthesis. We use supervised learning to model the mapping between motion features and sound parameters — the control parameters of a given synthesizer. Because demonstrations are composed of sequences of sound parameters (not the audio itself), the choice of the sound synthesis technique is essential. We previously proposed to use a graphical editors for specifying the sound parameter trajectories of the demonstration sounds, using physical modeling sound synthesis [34]. In this work, we focus on descriptor-driven concatenative sound synthesis techniques [78]. This particular case of sample-based sound synthesis allows to synthesize sounds from a set of audio descriptors that can be automatically extracted from audio recordings. Instead of designing sounds individually, which can be time-consuming, users can therefore create the demonstration sounds from existing collections of audio recordings. In this article, we also consider a case where users directly produce the demonstration sounds by vocalization.

Machine Learning Modeling of the Mapping. We use supervised learning techniques to estimate the mapping from a small set of multimodal demonstrations. This requires the use of methods that can efficiently learn from few examples.

Moreover, the training should be fast to allow users to rapidly iterate between demonstration and direct evaluation. The next section detail two probabilistic models of motion-sound relationships that can be used for jointly recognizing gestures and generating sequences of associated sound parameters in real-time.

5 PROBABILISTIC MODELING OF MOTION-SOUND RELATIONSHIPS

In this section, we detail two probabilistic models for real-time movement control of sound synthesis. Our approach to probabilistic mapping of movement and sound relies on the conversion between models with joint multimodal distribution to models with cross-modal conditional distributions. We detail detail regression methods that rely respectively on Gaussian Mixture Models and Hidden Markov Models.[2]

5.1 Gaussian Mixture Regression

Gaussian Mixture Regression (GMR) takes advantages of Gaussian Mixture Models (GMMs) for regression. The method was initially proposed by Ghahramani and Jordan for missing data estimation [39]. It has been applied to statistical speech synthesis [83, 84] as well as robot movement generation [13]. In this section, we briefly summarize the representation, learning and regression. A complete mathematical description of GMR can be found in [39, 79].

For training, we use multimodal feature vectors built by concatenating motion and sound feature vectors. We train a GMM on these features using the standard Expectation-Maximisation algorithm. For prediction, the weight of each gaussian component is estimated from the movement features only, and the associated sounds features are predicted by regression over each component.

5.1.1 Representation and Learning. Each demonstration is represented as a sequence of synchronized frames of motion features $\boldsymbol{x}^m$ and frames of sound parameters $\boldsymbol{x}^s$. For training, we consider the multimodal observation vectors built by concatenating motion and sound features $\boldsymbol{x} = [\boldsymbol{x}^m ; \boldsymbol{x}^s]$. We estimate a GMM of parameters $\boldsymbol{\theta}$ with K components, using the joint probability density function [66]:

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k\right) \tag{1}$$

where w_k is the weight of the kth component. $\mathcal{N}$ is a multivariate normal distribution, which mean $\boldsymbol{\mu}_k$ and covariance Σ_k can be expressed as a combination of the parameters for each modality (m for movement and s for sound). The mean of each Gaussian distribution is a concatenation of the mean for each modality, and the covariance matrix combines four submatrices representing uni-modal and cross-modal dependencies:

$$\boldsymbol{\mu}_k = \left[\boldsymbol{\mu}_k^m ; \boldsymbol{\mu}_k^s\right] \quad \text{and} \quad \Sigma_k = \begin{bmatrix} \Sigma_k^{mm} & \Sigma_k^{ms} \\ \Sigma_k^{sm} & \Sigma_k^{ss} \end{bmatrix} \tag{2}$$

For training, we use the Expectation-Maximization (EM) algorithm to estimate the mean, covariance, and weight of each component of the mixture. Algorithmic details on the EM algorithm can be found in [10, 66].

[2] In this section, we follow the mathematical conventions used in [66], and we refer the reader to chapters 11 and 17 for details on the standard algorithms for GMMs and HMMs.

5.1.2 Regression. For regression, our goal is to estimate the sound parameters $\boldsymbol{x}^s$ from input motion features $\boldsymbol{x}^m$. For this purpose, the joint density distribution must be converted to a conditional distribution that expresses the dependency of the sound modality over the input space of motion parameters. Following [39], the conditional distribution for a GMM can be expressed as:

$$p\left(\boldsymbol{x}^s \mid \boldsymbol{x}^m, \boldsymbol{\theta}\right) = \sum_{k=1}^{K} \beta_k \mathcal{N}\left(\boldsymbol{x}^s \mid \hat{\boldsymbol{\mu}}_k^s, \hat{\Sigma}_k^{ss}\right) \tag{3}$$

where

$$\begin{cases} \hat{\boldsymbol{\mu}}^s = \boldsymbol{\mu}^s + \Sigma^{sm} \left(\Sigma^{mm}\right)^{-1} \left(\boldsymbol{x}^m - \boldsymbol{\mu}^m\right) \\ \hat{\Sigma}^{ss} = \Sigma^{ss} - \Sigma^{sm} \left(\Sigma^{mm}\right)^{-1} \Sigma^{ms} \end{cases} \tag{4}$$

and where the responsibility β_k of component k over the space of motion features is defined by

$$\beta_k = \frac{w_k p\left(\boldsymbol{x}^m \mid \boldsymbol{\theta}_k\right)}{\sum_{k'} w_{k'} p\left(\boldsymbol{x}^m \mid \boldsymbol{\theta}_{k'}\right)} \quad \text{with} \quad p\left(\boldsymbol{x}^m \mid \boldsymbol{\theta}_k\right) = \mathcal{N}\left(\boldsymbol{x}^m \mid \boldsymbol{\mu}_k^m, \Sigma_k^{mm}\right) \tag{5}$$

We use the Least Square Estimate (LSE) to generate the vector of sound parameters from an input vector of motion features. The estimate can be computed as the conditional expectation of $\boldsymbol{x}^s$ given $\boldsymbol{x}^m$:

$$\hat{\boldsymbol{x}}^s = E\left[\boldsymbol{x}^s \mid \boldsymbol{x}^m, \boldsymbol{\theta}\right] = \sum_{k=1}^{K} \beta_k \hat{\boldsymbol{\mu}}_k^s \tag{6}$$

5.2 Hidden Markov Regression

We now consider an alternative to GMR that integrates a model of the gestures' time structure. Our method uses Hidden Markov Models for regression, which we denote Hidden Markov Regression (HMR). We start by learning a HMM on synchronous recordings of motion and sound parameters. Then, we convert the joint model to a conditional model: for each state, we express the distribution over sound parameters conditionally to the motion parameters. In performance, we use the input motion features both to generate the associated sound parameters from the conditional distribution. We propose an online estimation algorithm for HMR based on the Least Squares Estimate of the output sound parameters that allows for real-time parameter generation.

5.2.1 Representation and Learning. For training, we use a standard HMM representation with multimodal features. A HMM consists of a discrete-time, discrete-state Markov chain, with hidden states $z_t \in \{1 \cdots N\}$, plus an observation model $p(\boldsymbol{x}_t \mid z_t)$ [66] where $\boldsymbol{x}_t$ and z_t are the feature vector and hidden state at time t, respectively. The joint distribution of a HMM for the sequence of T feature vectors $\boldsymbol{x}_{1:T}$ and the state sequence $z_{1:T}$ can be written as:

$$p\left(\boldsymbol{x}_{1:T}, z_{1:T}\right) = p(z_{1:T}) p(\boldsymbol{x}_{1:T} | z_{1:T}) = \underbrace{\left[p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1})\right]}_{\text{(a) Markov process}} \underbrace{\left[\prod_{t=1}^{T} p(\boldsymbol{x}_t | z_t)\right]}_{\text{(b) observation model}} \tag{7}$$

where the Markov process is represented by the prior $\Pi = [\pi_i]_{i=1\cdots K}$ and the transition matrix $A = [a_{ij}]_{i,j=1\cdots K}$ so that $p(z_1 = i) = \pi_i$ and $p(z_t = j | z_{t-1} = i) = a_{ij}$. For continuous observations such as motion and sound features, we use a Gaussian observation model:[3]

$$p(\boldsymbol{x}_t \mid z_t = k, \boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{x}_t \mid \boldsymbol{\mu}_k, \Sigma_k\right) \tag{8}$$

[3] For simplicity, we considered a single Gaussian per observation distribution, however our implementation allows to define an observation model as a Gaussian mixture.

where the mean $\boldsymbol{\mu}_k$ and covariance Σ_k can be expressed with Equation 2. We use HMMs for modeling gestures composed of time series of motion and sounds features. A particular gesture is therefore modeled using a fixed number of hidden states that encode the temporal evolution of the features. To guarantee the consistency of the temporal process, we use a left-right topology for the transition matrix, which only allows transitions forward in time (See [71] for details). We train the HMM using the Baum-Welch algorithm [71] on a set of demonstrations (times series of joint motion and sound features), in order to estimate the transition probabilities and the mean and variance of each Gaussian observation model.

5.2.2 Regression. Several techniques for sequence mapping with Hidden Markov Models have been proposed for speech synthesis [21, 22, 38, 86]. However, most approaches in speech synthesis focus on the offline estimation of the output sequence. Our goal is to control the sound synthesis continuously and in real-time, which requires the generation of sound parameters as soon as a new frame of motion parameters is available. Our method is identical to that of Calinon et al. [14] for movement generation in robotics.

We aim to estimate $p(\boldsymbol{x}_t^s \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta})$, the distribution over the sound parameters at time t conditioned on the history of motion features up to time t:

$$\begin{aligned} p(\boldsymbol{x}_t^s \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta}) &= \sum_{i=1}^{N} p(\boldsymbol{x}_t^s, z_t = i \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} p(\boldsymbol{x}_t^s \mid \boldsymbol{x}_{1:t}^m, z_t = i, \boldsymbol{\theta}) p(z_t = i \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta}) \end{aligned} \tag{9}$$

Given that the observation model is Gaussian (see Equation 8), we can express the conditional observation model as:

$$p(\boldsymbol{x}_t^s \mid \boldsymbol{x}_{1:t}^m, z_t = i, \boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{x}_t^s \mid \hat{\boldsymbol{\mu}}_i^s, \hat{\Sigma}_i^{ss}\right) \tag{10}$$

where $\hat{\boldsymbol{\mu}}_i^s$ and $\hat{\Sigma}_i^{ss}$ can be expressed from $\boldsymbol{x}_t^m$, $\boldsymbol{\mu}_i$ and Σ_i using Equation 4. The filtered estimate of state probabilities $p(z_t = i \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta})$ can be computed in real-time using the forward algorithm [71]. The forward algorithm estimates the posterior likelihood of state z_t given the observation sequence $\boldsymbol{x}_{1:t}$ for a HMM with parameters $\boldsymbol{\theta}$ recursively:

$$\alpha_t^m(j) \triangleq p(z_t = j \mid \boldsymbol{x}_{1:t}^m, \theta) = \frac{\hat{\alpha}_t^m(j)}{\sum_{j'=1}^{N} \hat{\alpha}_t^m(j')} \tag{11}$$

where

$$\begin{aligned} \hat{\alpha}_t^m(j) &= \left[\sum_{i=1}^{N} \alpha_{t-1}^m(i) a_{ij}\right] p(\boldsymbol{x}_t^m | z_t = j, \theta) \\ &= \left[\sum_{i=1}^{N} \alpha_{t-1}^m(i) a_{ij}\right] \mathcal{N}\left(\boldsymbol{x}_t^m \mid \boldsymbol{\mu}_j^m, \Sigma_j^{mm}\right) \end{aligned} \tag{12}$$

The algorithm is initialized as follows:

$$\alpha_0(i) = \frac{\pi_i \mathcal{N}\left(\boldsymbol{x}_0^m \mid \boldsymbol{\mu}_i^m, \Sigma_i^{mm}\right)}{\sum_{j=1}^{N} \pi_j \mathcal{N}\left(\boldsymbol{x}_0^m \mid \boldsymbol{\mu}_j^m, \Sigma_j^{mm}\right)} \tag{13}$$

Similarly to GMR, we use the Least Square Estimate (LSE) for generating the sound parameters associated to an input frame of motion features. Formally, the sound parameter vector can therefore be expressed as:

$$\hat{\boldsymbol{x}}_t^s = E\left[\boldsymbol{x}_t^s \mid \boldsymbol{x}_{1:t}^m, \boldsymbol{\theta}\right] = \sum_{i=1}^{N} \alpha_t^m(i)\hat{\boldsymbol{\mu}}_i^s \tag{14}$$

5.3 Implementation for User-centered Design

Making machine learning usable for users and interaction designers requires models that can be trained from few examples. Setting the model parameters to appropriate values is therefore essential. Many model selection methods have been proposed in the machine learning literature for automatically selecting optimal parameters. However, as shown by Fiebrink et al. for the case of musical instrument design [28], users often prefer to evaluate models by direct interaction rather than through metrics such as the classification accuracy. Fiebrink reported that in some cases, the classification boundary is more relevant to users because it informs them on the amount of variations of the gestures that are meant to be recognized. We consider that two parameters of the proposed probabilistic models are essential for interaction design: complexity of the model and its regularization. This section aims to give practical insights supporting the choice of particular models, and the adjustment of their parameters.

5.3.1 Choice of GMR vs HMR. GMR and HMR have different properties for modeling the mapping between motion and sound. The main difference is that HMR has an explicit model of the temporal evolution of the features, while GMR does not take time into account. The choice of a model should be determined according to the use-case, by analyzing the type of gestures and of relationships between motion and sound.

GMR is particularly appropriate to design continuous mappings between motion and sound that have a one-to-one relationship between particular values of motion and sound parameters. In some cases, however, there can be ambiguities in the feature spaces that result in one-to-many associations between values of motion and sound parameters. In this case, the temporal model of HMR can help resolving the ambiguities because it intrinsically takes into account the history of the movement features. Such ambiguities can also arise from a choice of sensors: acceleration signals can present specific temporal patterns that make difficult to associate particular acceleration values to sound parameters with a one-to-one mapping. However, the constraints imposed by the temporal structure of HMR can also limit the possibility for extrapolation.

5.3.2 User Control of the Model Complexity. Model complexity can be handled through the number of Gaussian components in GMR, or the number of hidden states in HMR. The non-linearity of the mapping increases with the number of state or components. Using a small number of hidden states implies that the information of the movement is embedded in a lower dimensional space, reducing the accuracy of the temporal modeling of the gesture. Using few states can help ensuring a good generalization of the model. The recognition will therefore be tolerant to variations in the input, which might help when working with novice users, or when the end users do not design the gestures themselves. Conversely, choosing a large number of states — relatively to the average duration of the training examples, — increases the accuracy of the temporal structure. This can help ensure that the temporal structure of the synthesized sound parameters is coherent with the demonstrations. It can also be useful for expert users, such as musicians, who can repeat gestures with high consistency.

5.3.3 Regularization. Regularization is essential when training machine learning models from few examples. It can prevents numerical errors during training by avoiding that variances tend towards zero. More importantly, it also allows users to control the degree of generalization of the model when the training set is too small to ensure a robust estimation of the data covariance. For GMR and HMR, regularization artificially increases the variance and overlap of the Gaussian components, which impacts the smoothness of the generated sound trajectories.

We implemented regularization through a prior σ added to the covariance matrices of the Gaussian distributions at each re-estimation in the Expectation-Maximization algorithm. Our regularization method is a special case of the Bayesian regularization technique proposed by [68]. σ combines an absolute prior and a relative prior:

- *Absolute Regularization* σ^{abs} represents the absolute minimal value to add to the diagonal of the covariance matrix.
- *Relative Regularization* σ^{rel} is proportional to the standard deviation of each feature on the entire training set.

At each iteration of the Expectation-Maximization (EM) algorithm, we estimate the regularized covariance matrix $\tilde{\Sigma}$ from the covariance matrix Σ estimated via EM as

$$\tilde{\Sigma}_{ii} = \Sigma_{ii} + \max\left(\sigma^{rel} * \sigma_i, \sigma^{abs}\right) \quad \forall i \in [1; D] \tag{15}$$

where D is the total number of features, Σ_{ii} is the ith value of the diagonal of the covariance matrix Σ and σ_i represents the standard deviation of feature i on the entire training set.

5.3.4 Joint Recognition and Mapping. Our implementation gives users the possibility to define ‘classes’ of motion-sound mappings, by annotating demonstrations with discrete labels. This process allows for joint recognition and regression: in *performance*, we use the movement to jointly estimate the likelihood of each class and their associated sound parameters. In practice, we train one model (GMR or HMR) per class,[4] using all demonstrations with a given label. During performance, we evaluate the posterior likelihood of each class given the motion features. The likeliest class is then used to generate the sound parameters.[5]

5.3.5 The XMM Library. We released our implementation of GMR and HMR as an open-source library for continuous motion recognition and mapping. XMM[6] is a portable, cross-platform C++ library that implements Gaussian Mixture Models and Hidden Markov Models for recognition and regression. XMM has python and Node.js bindings for scripting and web integration. The models are also integrated with the *MuBu*[7] environment within *Cycling 74 Max*. It provides a consistent framework for motion/sound feature extraction and pre-processing; interactive recording, editing, and annotation of the training sets; and interactive sound synthesis. This set of tools reinforces the fluidity of the workflow for recording, training and evaluating of the models, and testing the mapping.

6 CASE STUDY: INTERACTIVE AUDIO FEEDBACK USING VOCALIZATIONS

We now detail the case study of a particular implementation of a mapping through interaction system. In this project, we considered voice and gestures as primary material for interaction design. Humans use vocalizations in conjunction with gestures in a number of situations, from everyday communication to expert movement performance. Vocalization is an efficient and expressive way to provide the system with sound examples that can be accurately performed in synchrony with movements. This project is motivated by the extensive use of (non-verbal) vocalization as a support to

[4]Note that our multiclass implementation of HMR allows the use of a hierarchical extension of Hidden Markov Models, that we previously used for real-time gesture recognition [32].
[5]Note that our implementation also allows to generate the sound parameters as a mixture of the prediction of each class, where the likelihoods are used as mixing weights.
[6]The XMM open-source library: https://github.com/Ircam-RnD/xmm
[7]MuBu is freely available on Ircam’s *Forumnet*: http://forumnet.ircam.fr/product/mubu/

movement expression in dance practice and pedagogy [61] as well as in dance movement therapy [5]. Similarly, expert choreographers integrate vocal sounds in rehearsal and practice to communicate choreographic ideas to performers [55]: timing and rhythm, but also movement dynamics and 'quality', or even imagery [89]. We were interested in exploring how novice participants would associate particular gesture with sounds produced vocally. Our system relies on jointly performed gestures and vocalizations to train a joint multimodal model that encode the time-evolving dynamics of movement and sound. It enables users to continuously control the synthesis of vocalizations from continuous movements.

6.1 System Description

The architecture of the application is outlined in Figure 2.[8] Users build the demonstrations by producing a vocalization while performing a gesture. We capture the movements using body-worn inertial sensors. The system uses HMR to learn the relationship between a sequence of motion parameters and a sequence of Mel-Frequency Cepstrum Coefficients (MFCCs) representing the vocal sound. Once the model is learned, users can perform new movements to control the sound synthesis. Motion parameters are streamed to the trained HMR that predicts MFCCs for each new input frame. The MFCC are then used to synthesize sound based on the vocalization, using descriptor-driven granular synthesis.

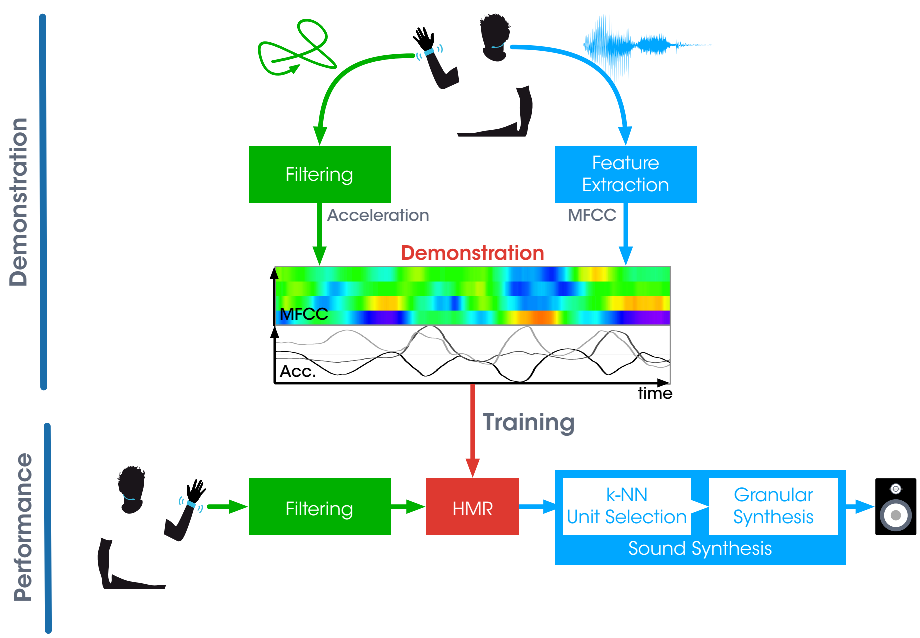


Fig. 2. Overview of the interactive vocalization system. The demonstrations are built by the player who co-produce a gesture and a vocalization. The mapping is modeled by Hidden Markov Regression, and vocal sounds are resynthesized using descriptor-driven granular synthesis.

Motion Capture. Users are equipped with an Inertial Measurement Unit (IMU) strapped on the right wrist. The IMU includes a 3D accelerometer and a 3-axis gyroscope (as IMU we used the Modular Musical Object (MO) described

[8] A video demonstrating the system for gesture-based control of vocalizations can be found online: http:vimeo.com/julesfrancoise/mad

in [72]). Data frames are streamed to the computer running the software using the Zigbee protocol at a fixed sampling rate of 100 Hz.

Motion Representation. The motion is represented using a set of low-level features from the inertial data. We use a concatenation of the acceleration data, the gyroscopic data, and the first derivative of the acceleration data, for a total of 9 dimensions. The data is smoothed using a moving-average filter with a window size of 6 frames. To ensure a good signal-to-noise ratio, the derivative is computed on a low-pass filtered version of the acceleration. These parameters were fine-tuned empirically along an iterative design process.

Audio Analysis. Vocalizations are recorded at 44.1 kHz using a microphone. We use the PiPo library to extract 12 Mel-Frequency Cepstral Coefficients (MFCCs), with a window size of 46.4 ms and a hop size of 11.6 ms. MFCCs are then resampled at 100 Hz to match the sampling rate of the motion data, and smoothed using a moving-average filter with a window size of 6 frames.

Mapping. Motion parameters, audio, and MFCCs are synchronously recorded using MuBu. we use the MuBu implementation XMM library to train a HMR model for each vocalization. The HMM can be parametrized manually. Based on informal iterative testing, we consider that using 20 states per gesture and a relative regularization of 0.2 provide high quality feedback and allow for rapid training. To generate the sound feedback, the motion parameters are streamed to the trained HMR that predicts, for each input frame, the associated MFCC parameters.

Sound Synthesis. Vocalizations are resynthesized in real-time using descriptor-driven granular synthesis. Our synthesis method is similar to concatenative sound synthesis [78], with shorter sound segments. From a frame of MFCCs, we estimate the position within the original vocalization of the nearest sound segment — .i.e with the smallest distance between MFCCs. The estimated temporal position is then used to drive a granular synthesis engine with a grain duration of 100 ms, 90% overlap, and a 3ms random variation of the temporal position.

6.2 Presentation as a Public Installation: The 'Imitation Game'

We created an imitation game that engages two players in imitating each other's gestures to reproduce vocal imitations (shown at For SIGGRAPH'14 Emerging Technologies [35, 36]]. The setup of the game is illustrated in Figure 3. The game started with a participant recording a vocal imitation along with a particular gesture. Once recorded, the systems learned the mapping between the gesture and the vocalization, allowing users to synthesize the vocal sounds from new movements. The second player then had to mimic the first player's gesture as accurately as possible to resynthesize the vocalization accurately and win the game.

To support the creation of gestures and vocalizations, we created a set of 16 *action cards* that gave a pictorial representation of an action with its associated sound as a phonetic indication (see Figure 4). To prepare the game, each player recorded several vocal and gestural imitations. During the playing phase, participants were shown sequences of the action cards. Participants had to reproduce the gestures, remaining as accurate as possible while the pace was accelerating, allocating less time to imitate each gesture. Along the game, participants won points according to a measure of how accurately they were able to reproduce the gestures' dynamics. This measure was based on the likelihood computed by the models for particular gestures, and allowed us to analyze the performance of the participants over time. Statistical analysis showed that participants with higher expertise (i.e. the demonstrators) had significantly higher likelihoods in average, and that the likelihood for all participants decreased as the game accelerated. This acceleration was leading participants to reproduce the gestures faster and with more energy, which eventually lead to decreased

Fig. 3. The *imitation game*: the players use interactive audio feedback to improve their imitations of each other's gestures. A set of action cards support the creation of particular metaphors.

likelihood with respect to the original demonstration. Nonetheless, the gamification of the system increased participants' engagement with the installation.

Qualitative observations support the idea that interactive sound feedback helps reproducing gestures characteristics that are not easily apprehended using the visual modality only. Often, participants managed to reproduce the dynamics of the demonstrator's gesture by iteratively exploring the movement and its relationship to the sound feedback. We hypothesize that combining the auditory feedback with verbal guidance allowed to quickly converge to the correct motion dynamics. In many cases, we observed a quick adaptation of the participants along the trials.

6.3 Applications in Music and Dance

Composer Greg Beller used both our systems based on Hidden Markov Regression and Gaussian Mixture Regression in his contemporary performance project "Synekine" during his musical research residency at Ircam. In the Synekine project, "the performers develop a fusional language involving voice, hand gestures and physical movement. This language is augmented by an interactive environment made of sensors and other Human-Computer Interfaces." [4].

We also applied the system to movement quality sonification for dance pedagogy. In Laban Movement Analysis, vocalization is used to support the performance of particular Efforts relating to movement qualities. We conducted a study on the sonification of Laban Effort factors using the vocalization system [33]. We trained a system using expert performances of vocalized movement qualities, that we used in an exploratory workshop to support the pedagogy of Laban Effort Factors with dancers. Finally, we used the system as a tool for sketching interaction in sonic interaction design, in the framework of the SkAT-VG European project [73].

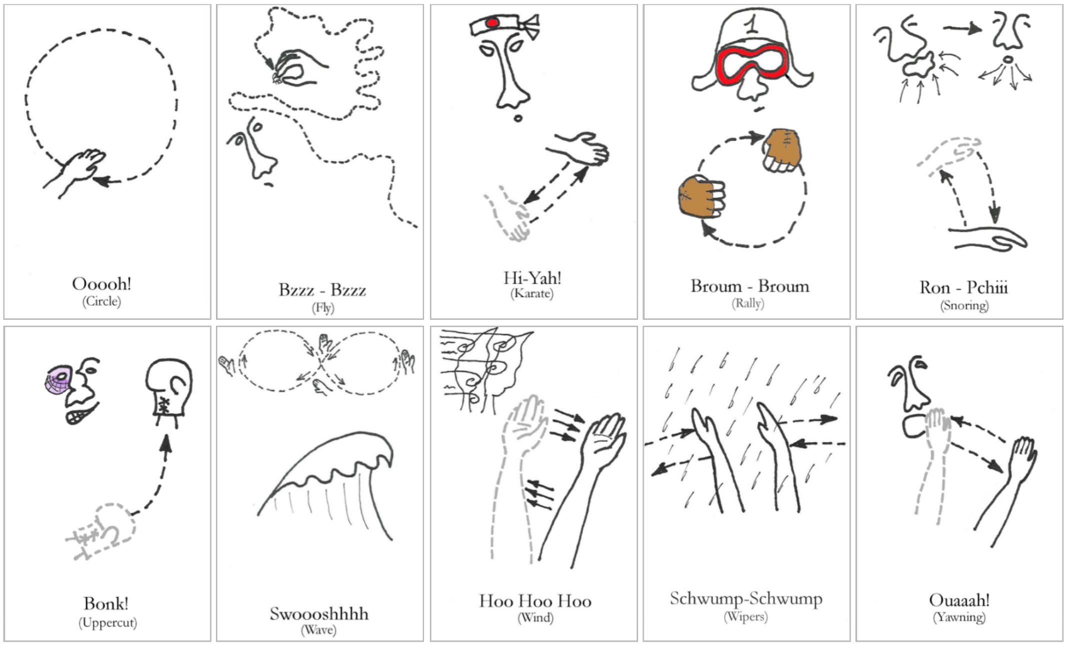


Fig. 4. Selection of 10 of the action cards designed for the imitation game. Each card depicts a metaphor involving both a gesture and a vocalization (cards designed and drawn by R. Borghesi).

7 EVALUATION

We focus on machine learning as a tool for user-centered design. Mapping through Interaction lets users define their own gestures and sounds, and it is often the same person that defines the training data to later peform the interactive system. As a result, we focus on evaluating the different models on user-defined gestures, where models are trained and evaluated on gestures from the same user.

In order to evaluate the proposed method of Hidden Markov Regression, we conducted a study using the interactive vocalization system (section 6.1). We captured a dataset of gestures co-produced with vocalizations, in order to assess the quality of the synthesized sound parameters from an input gesture with various models: Hidden Markov Regression (HMR), Gaussian Mixture Regression (GMR), and a set of standard regression methods. We selected a set of models readily available within the scikit-learn machine learning toolbox [70]: Support Vector Regression (SVR), Multi-Layer Perceptron (MLP) and Gaussian Process Regression (GPR). A full description of each of these models is beyond the scope of the current paper, details and references can be found in the scikit-learn documentation.

7.1 Protocol

We designed 10 gestures and their associated vocalizations, based on the set of action cards created for the 'imitation game' introduced in the previous section. Participants were asked to record a set of executions of each gesture, alternating between (1) co-producing a gesture and a vocalization, and (2) executing the gesture with the interactive audio feedback.

7.1.1 Participants. We recruited 10 participants (6 women, 4 men), aged from 24 to 42 (mean=32.4, SD=6.3). Participants did not have previous expertise with the proposed system. The experiment was exclusively performed with the right hand, and 1 participant was left-handed.

7.1.2 Apparatus. Participants were sitting on a chair in front of the computer running the software. They were equipped with a motion sensor and their vocalizations were recorded using a microphone adjusted to their position. The software ran on an Apple MacBook Pro with a 2.9 GHz Intel core i5 processor and 8GB memory. The software and interface were implemented using *Cycling'74 Max* and the *MuBu for max* library[9] for motion analysis and sound synthesis. We used the system previously described, with an Audio Technica ATM31a microphone. For the parts with audio feedback, the HMR was parametrized with 20 states and a relative regularization of 0.2, based on the configuration used for the public installation.

Participants were presented with an interface composed of two panels. The top panel presented to the participant both the 'action card' — with a schematic description of the gesture and a text description of the vocalization, — and a video of one of the experimenters showing the gesture and vocalization to reproduce.[10] The bottom panel allowed participants to start and stop the recordings, and informed them wether to perform both the gesture and vocalization, or only the gesture with the interactive audio feedback.

7.1.3 Procedure. Participants were asked to record a total of 15 executions of the 10 gestures, raising a total of 150 recordings per participant. Gestures were designed from the action cards created for the imitation game, as depicted in Figure 4. Actions cards were presented sequentially, and their order was randomized for each participant, in order to avoid any order effect. The procedure for each gesture, depicted in Figure 5, was as follows:

(1) Observe the action card and the associated video example. There was no limitation on the number of viewings, and participants were invited to imitate the gesture and vocalizations displayed in the video
(2) Perform 5 iterations of 3 recordings:
    Demo Record a gesture co-produced with a vocalization
    Exec Record two executions of the gesture only, with the interactive audio feedback trained on the previous gesture-voice recording.

Participants were instructed to remain as consistent as possible between recordings, and that the quality of the audio feedback depended on the consistency of their gesture execution with regards to the demonstration.

7.1.4 Data Collection. The resulting dataset contains 500 demonstrations (coproduced gestures and vocalizations), and 1000 gestures performed with the interactive audio feedback, from a total of 10 participants. Each execution contains synchronized recordings of:

- The motion features and raw IMU data
- The audio, either from the microphone during vocalization, or from the sound synthesis during executions with feedback.
- The audio description as MFCCs, either computed from the microphone during vocalization, or predicted by the HMR model during executions with feedback
- A video of the participant from the laptop's built-in camera, for control.

[9] http://forumnet.ircam.fr/product/mubu/
[10] The example videos are available online as supplementary material: https://www.julesfrancoise.com/tiis-supplementary/

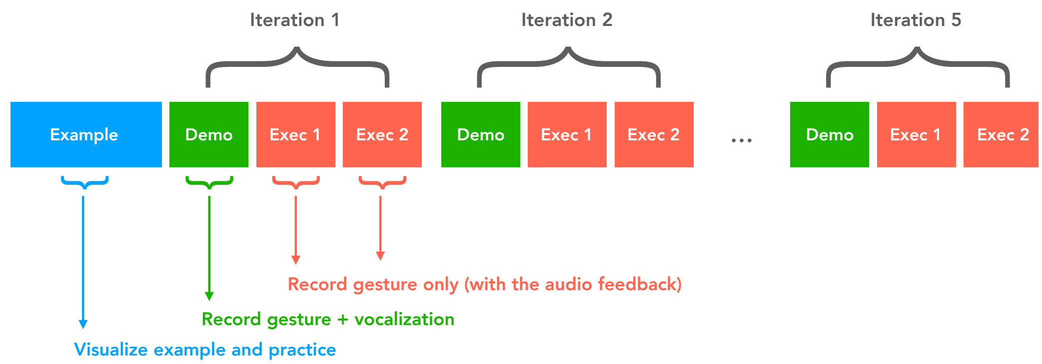


Fig. 5. Overview of the procedure for each gesture. Participants start by observing the action card and the example video recording. When ready, they proceed to 5 iterations of recordings, each composed of one demonstration (gesture and vocalization are co-produced) and two executions (only the gesture is performed) with the audio feedback.

7.2 Evaluation of the Regression Error

Mapping through Interaction aims to enable users to prototype through rapid iterations. It requires that regression models learn efficiently from few examples, and that the output parameters are generated in real-time. Our evaluation focuses on one-shot learning (from a single demonstration) of user-defined gestures and vocalizations. Training and prediction are therefore performed on data from the same participant. We evaluate the error between the sequence of MFCCs predicted causally by a model from a user's gesture and the true vocalization of the user. We compare Hidden Markov Regression (HMR), Gaussian Mixture Regression (GMR), Support Vector Regression (SVR), Gaussian Process Regression (GPR) and the Multi-Layer Perceptron (MLP). We used the XMM implementation of HMR and GMR, and the python Scikit-Learn[11] library for SVR, GPR and MLP.

We report the results of two evaluation of the regression error. The first evaluation focuses on pure regression where the same gesture is used for training and testing. The second evaluation considers a multi-class problem involving joint recognition and regression, where models are trained using 10 demonstrations associated with different action cards.

7.2.1 Single-class Regression. We measure the regression error of each model according to the following procedure. For each gesture, each participant, a model is trained on one demonstration. For all other demonstrations of the same gesture by the same participant, we use the trained model to predict the sequence of MFCC of the associated vocalization. The regression error is measured as the mean squared error between the predicted MFCC and the MFCC of the true vocalization.

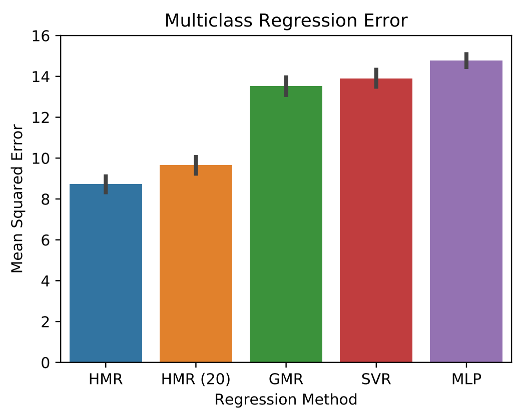
We optimized the parameters of each model by grid search on a subset of the dataset composed of the gestures of 2 participants. HMR was configured with $N = 50$ hidden states and a relative regularization $\sigma^{rel} = 2$. GMR was configured with $N = 20$ Gaussians components and a relative regularization $\sigma^{rel} = 0.2$. SVR used a Radial Basis Function (RBF) kernel of coefficient $\gamma = 1.0$, with a penalty of the error term $C = 1$. GPR was optimized by stochastic gradient descent, used a Radial Basis Function (RBF) kernel, and a regularization $\alpha = 1e^3$. MLP was configured with a single layer of 1000 hidden units with logistic activation, regularization (L2-penalty) $\alpha = 0.1$. We additionally report the results of Hidden Markov Regression with $N = 20$ hidden states and a relative regularization $\sigma^{rel} = 0.2$, that was used

[11] http://scikit-learn.org/

to provide feedback to the participants during the study. We used the demonstrations from the 8 remaining participants for testing.

Figure 6a reports the mean squared regression error for a single-class problem for all models. In order to test for statistical significance of the observed differences in mean, we computed an ANOVA with post-hoc Tukey paired tests, after checking for normality and homogeneity of variances. With one-way repeated-measures ANOVA, we found a significant effect of the model on the regression error ($F(5, 9714) = 231$, $p < 0.001$, partial $\eta^2 = 0.12$). A Tukey's pairwise comparison revealed the significant differences between all models ($p < 0.01$). The results indicate that HMR performs significantly better than all other regression methods with parameters set to optimal values. The regression error for GMR is also significantly lower than for other methods, which shows that our implementation can efficiently learn from few examples on a dataset of user-defined gestures and vocalizations. While our parameter choice for the user study (20 states, $\sigma = 0.2$) is not optimal, it still outperforms other models. It also reduces the training time, which is essential for rapid prototyping. The performance of the algorithm was consistent across the various action cards, which confirms the performance of the method for a number of different gestures.

(a) Mean squared regression error for a single-class setting (same action card used for training and testing) using various regression methods. Results present the mean and 95% confidence interval for each method. Models used the following configuration: 'HMR': Hidden Markov Regression ($N = 50$, $\sigma^{rel} = 0.7$), 'HMR (20)': Hidden Markov Regression ($N = 20$, $\sigma^{rel} = 0.2$), 'GMR': Gaussian Mixture Regression ($N = 20$, $\sigma^{rel} = 0.05$), 'SVR': Support Vector Regression (RBF Kernel, $C = 1$, $\alpha = 1$), 'GPR': Gaussian Process Regression (RBF Kernel, $\alpha = 1e3$), 'MLP': Multi-Layer Perceptron (Logistic activation, $N = 1000$, $\alpha = 0.1$)



(b) Mean squared regression error for a multi-class setting (10 action cards used for training) using various regression methods. Results present the mean and 95% confidence interval for each method. Models used the following configuration: 'HMR': Hidden Markov Regression ($N = 50$, $\sigma^{rel} = 0.7$), 'HMR (20)': Hidden Markov Regression ($N = 20$, $\sigma^{rel} = 0.2$), 'GMR': Gaussian Mixture Regression ($N = 20$, $\sigma^{rel} = 0.05$), 'SVR': Support Vector Regression (RBF Kernel, C=1, $\alpha = 1$), 'MLP': Multi-Layer Perceptron (Logistic activation, $N = 1000$, $\alpha = 0.1$)

Fig. 6. Regression error in single-class and multi-class settings.

7.2.2 Multi-class Joint Recognition and Regression. We now consider a regression problem with multiple classes of gestures. Each class is represented by an action card. Our goal is to let users define continuous mappings for each action card, and to jointly perform the recognition of the card and the generation of sound parameters. For each gesture, each participant, a model is trained on one demonstration of each of the 10 action cards, annotated with its label. For all other demonstrations of the same gesture by the same participant, we use the trained model to predict the sequence of MFCC of the associated vocalization. The regression error is measured as the mean squared error between the predicted MFCC and the MFCC of the true vocalization.

As before, we optimized the parameters of each model by grid search on a subset of the dataset composed of the demonstrations of 2 participants. For HMR and GMR, we trained 10 models for each of the labels. For prediction, we evaluate at each frame the likeliest model, which is used to predict the associated MFCCs. SVR and MLP do not intrinsically allow for the definition of classes and were trained on all frames of the 10 gestures without any label. HMR was configured with 50 hidden states and a relative regularization $\sigma^{rel} = 0.7$. GMR was configured with 30 Gaussians components and a relative regularization $\sigma^{rel} = 0.05$. SVR used a Radial Basis Function (RBF) kernel of coefficient $\gamma = 1.0$, with a penalty of the error term $C = 1$. GPR was optimized by stochastic gradient descent, used a Radial Basis Function (RBF) kernel, and a regularization $\alpha = 1e^3$. MLP was configured with a single layer of 1000 hidden units with logistic activation, regularization (L2-penalty) $\alpha = 0.1$. We additionally report the results of Hidden Markov Regression with 20 hidden states and a relative regularization $\sigma^{rel} = 0.2$, that was used to provide feedback to the participants during the study. We used the demonstrations from the 8 remaining participants for testing.

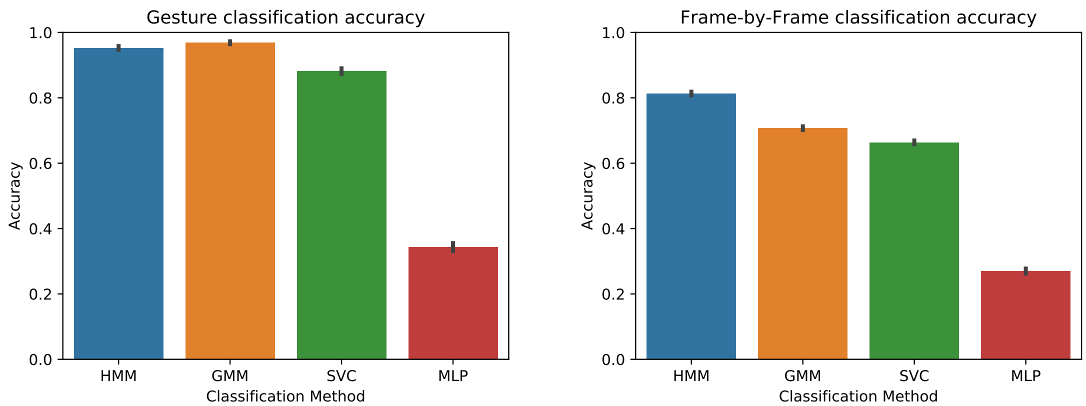
Figure 6b reports the mean squared regression error for a multi-class joint regression problem for all models. With one-way repeated-measures ANOVA, we found a significant effect of the model on the regression error ($F(5, 9714) = 231$, $p < 0.001$, partial $\eta^2 = 0.12$). A Tukey's pairwise comparison revealed the significant differences between all models ($p < 0.01$). The results indicate a similar trend in the performance of the various models. It is interesting to notice that the performance gain of HMR compared to other methods is larger than in the case of single-class regression. In particular, HMR performed significantly better than GMR, which shows that the temporal model introduced by the Markov process is critical for real-time recognition and mapping on a large set of gestures. In future work, we will compare the proposed approach with recurrent neural networks, that can integrate temporal dependencies at various time scales. Since there exist many forms and possible implementations of recurrent neural networks, an exhaustive evaluation is beyond the scope of this paper, and we focused on models with standard implementations.

7.3 Real-time Gesture Recognition

We also evaluated the ability of each model to recognize in real-time the various gestures labeled by action card. We measure the classification accuracy of user-defined gestures for each model according to the following procedure. For each participant, a model is trained from a single demonstration of each of the gestures. We then use the trained model to recognize the label of the action card of all other gesture recordings. We report to measures of accuracy: the classification of the entire gesture (one label per gesture), and the frame-by-frame classification (accuracy averaged on all frames of a gesture). Our method focuses on continuous recognition, where the likelihood of various classes is re-estimated at each new frame. This process is appropriate for real-time control of audio processing recognition: it enables to avoid any segmentation before the classification process, which might lead to inconsistent results with errors in the segmentation. For this reason, we report the frame-by-frame recognition rate that accounts for the model's ability to recognize the gestures in a continuous stream of movement data.

Analogously to the previous section, we compare the following models: Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), Support Vector Machines Classification (SVC) and the Multi-Layer Perceptron (MLP). We optimized the parameters of each model by grid search on a subset of the dataset composed of the gestures of 2 participants. HMMs were configured with $N = 10$ hidden states per class and a relative regularization $\sigma^{rel} = 0.1$. GMMs were configured with $N = 10$ Gaussians components per class and a relative regularization $\sigma^{rel} = 0.1$. SVC used a Radial Basis Function (RBF) kernel of coefficient $\gamma = 1$, with a penalty of the error term $C = 1$. MLP was configured with a single layer of $N = 500$ hidden units with logistic activation, regularization (L2-penalty) $\alpha = 0.1$. We used the demonstrations from the 8 remaining participants for testing.

Figure 7a reports the accuracy of each model for the classification of entire gestures. With one-way repeated-measures ANOVA, we found a significant effect of the model on the classification accuracy ($F(3, 22396) = 4913$, $p < 0.001$, partial $\eta^2 = 0.39$). A Tukey's pairwise comparison revealed the significant differences between all models ($p < 0.01$), except between HMMs and GMMs which show the highest average accuracy of 0.95 and 0.96 respectively.



(a) Classification of complete gestures (one label per gesture) (b) Frame-by-frame classification accuracy (one label per frame)

Fig. 7. Classification accuracy for personal gestures for various classifiers. 'hmm': Hidden Markov Regression ($N = 10$, $\sigma = 0.1$), 'gmm': Gaussian Mixture Regression ($N = 10$, $\sigma = 0.1$), 'svc': Support Vector Classification (RBF Kernel, $C = 1$, $\alpha = 1$), 'mlp': Multi-Layer Perceptron (Logistic activation, $N = 500$, $\alpha = 0.1$)

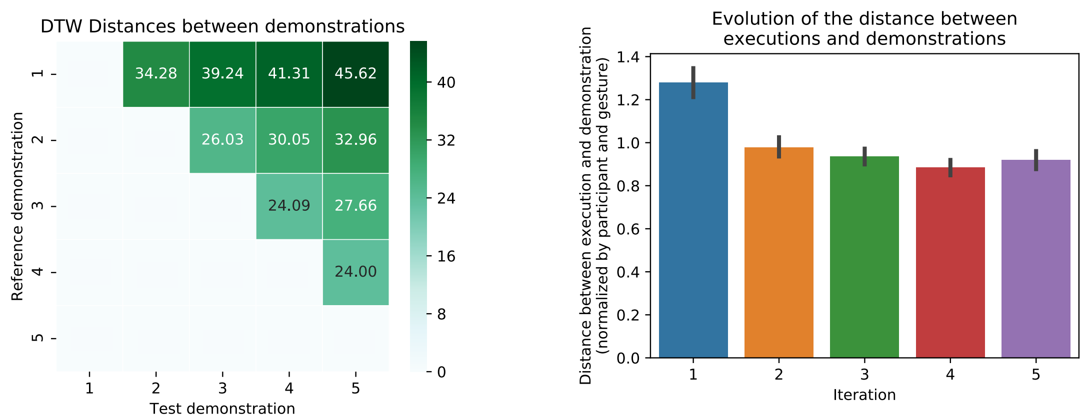
Figure 7b presents the accuracy of each model for frame-by-frame recognition. This corresponds to a desired use-case in music interaction, where the goal is to continuously recognize the likeliest class at each frame, to allow for low-latency sound control. For each gesture, we evaluate the accuracy as the average accuracy of the recognition at each frame. In other words, this measure represent the proportion of frames accurately classified over the entire gesture. With one-way repeated-measures ANOVA, we found a significant effect of the model on the classification accuracy ($F(3, 22396) = 5072$, $p < 0.001$, partial $\eta^2 = 0.40$). A Tukey's pairwise comparison revealed significant differences between all models ($p < 0.01$). HMMs have the highest accuracy, with an average of 80% of frames accurately recognized in each test gesture. Once again, the temporal model introduced by the Markov process is beneficial to the continuous recognition of user-defined gestures.

7.4 Evolution of Gesture Execution over Time

We now investigate participants' execution of the demonstrations and the gestures performed with audio feedback. Our goal is to evaluate whether participant's iteratively refined their demonstrations over the 5 iterations of demonstration for each gesture. We also assess the evolution over time of the consistency in their execution of gestures, where consistency is measured in terms of distance between gestures. We computed distances between recordings to assess the similarity between two executions of the same gesture. We define the distance between executions as the Dynamic

Time Warping (DTW) [53] distance between two sequences of motion features. DTW realigns the sequences and allows to alleviate the variations in timing for the different recordings. DTW is therefore advantageous to limit the influence of irrelevant errors introduced by the gesture segmentation — for instance, the duration between the start of the recording and the actual start of the participant's gesture can vary across recordings). We consider that the DTW distance relates to the consistency of movement execution: two

7.4.1 Evolution of the Demonstrations across Iterations. We computed the set of distances between the 5 demonstrations that users recorded for each action card. For each participant, each action card, we computed the DTW distance between the motion feature sequence of the demonstration of two iterations. Figure 8a reports the average distances between the first and last demonstration, and all other demonstrations. In the following, we denote the distance between demonstrations i and j as Dij.



(a) Evolution of the distance between demonstrations.

(b) Evolution of the DTW distance between the two executions and the demonstration of the same iteration

Fig. 8. Evolution of the distances between different recordings of the same gestures. Distances are computed as the DTW between two sequences of motion features from two executions of a gesture.

We observe that the distance to the first demonstration gradually increases at each iteration. With one-way repeated-measures ANOVA, we found a significant effect of the iteration on the distance to the first demonstration, although with a small effect size ($F(3, 396) = 3.8$, $p < 0.05$, partial $\eta^2 = 0.03$). A Tukey's pairwise comparison revealed a significant difference between D12 and D15 ($p < 0.01$). This means that although participants were instructed to remain consistent across iteration, the variations in the gesture execution are not random but highlight an evolution of the gesture over time. Similarly, the distance to the 5th demonstration diminishes along iterations.

The most significant result relates to the comparison of the distances to the first and last demonstration: $D1x$ and $D5x$. With a paired t-test, we found no significant difference between D12 and D52, meaning that the second demonstration is as similar to the first than the last demonstration. With paired t-tests, we found significant differences between D13 and D53, and between D14 and D54 ($p < 0.001$). This means that participants' gestures rapidly diverge from the first recording, and get increasingly stable over time.

7.4.2 Evolution of the Consistency between Demonstration and Executions across Iterations. We now investigate participants consistency over time, i.e. their ability to execute the gesture with the auditory feedback consistently with their demonstration. For each of the 5 iterations, we computed the distances between the executions performed with sound feedback, and the associated demonstration gesture. Because participants execute each gesture in a different way and because their baseline consistency varies, we normalized the distances by dividing each distance by the average distance for the same participant, same gesture.

Figure 8b reports the evolution over time of the normalized distance between the executions performed with sound feedback and their associated demonstration. A Kruskal-Wallis test revealed a significant effect of the iteration on the distance ($F(4) = 122.0$, $p < 0.001$). A post-hoc test using Mann-Whitney U tests with Bonferroni correction showed the significant differences of consistency between Demonstrations 1 and all other Demonstrations ($p < 0.001$, $r > 0.5$) and between Demonstration 2 and Demonstration 4 ($p < 0.01$, $r = 0.2$). While participants show a large variability in their execution on the first iteration of the process, results show that participants are rapidly improving the way they execute the gesture. The consistency keeps improving after the second iteration, though to a smaller extent. Note that the same distances have been computed on the MFCCs representing the vocalization, with analogous results.

8 DISCUSSION

Mapping through Interaction requires that all training examples are provided by the user, to allow for rapid personalization of the relationships between motion and sound. In this section, we discuss some of the critical aspects of the method regarding the challenges of generalization from few examples, user expertise, and the importance of the iterative design process.

8.1 Generalizing from Few Examples

Our study highlighted that the proposed methods of HMR and GMR outperform standard regression techniques on a dataset of user-defined gestures and vocalizations. It is interesting to notice that the optimal parameters for real-time joint recognition and generation involve large values of regularization. This shows that regularization is essential for one-shot learning, because it allows to better take into account the variations in a new execution of the gesture. Furthermore, we found that HMR outperformed all other techniques on this task and dataset, with an optimal configuration involving 50 hidden states. Using such a large number of states can result in overfitting. However, for user-specific gestures this large number of states increases the resolution of the time structure of the gesture, that helps improving the quality of the synthesis of sound parameter trajectories.

Our dataset was built to evaluate the prediction error for user-specific gesture control of sound feedback. Results show that the proposed method enables to learn individual differences in the execution of gestures and vocalizations, that greatly vary across action cards and participants. Further evaluating the extrapolation and interpolation between gestures would require the creation of a dataset of gestures and sounds with well-defined variations.

In our experience, HMR is advantageous for gestures that have a well-defined temporal evolution. In this case, HMR can extrapolate from a set of demonstrations with variations of the motion-sound relationship, as long as these variations follow a similar time structure. For continuous mappings that have a one-to-one correspondence between the motion and sound parameter spaces, GMR can extrapolate more consistently than HMR, and can be advantageous for multidimensional control of parametric synthesis.

8.2 Dealing with User Expertise

Users' ability to provide high quality demonstration is another essential condition for efficient learning from few example, and has been discussed in related areas such as robot programming-by-demonstration [3]. Through our user study, we found out that the regression error is significantly higher when the model is trained on the first demonstration rather than on a subsequent demonstration. Participants were able to rapidly adapt their gestures to make a more efficient demonstration, that would allow for more accurate resynthesis of the vocalization-based feedback. This observation highlights that human learning is necessary to efficient and expressive design with machine learning. Users can acquire expertise in the sensori-motor execution of particular gestures for a given task, but they can also learn at a longer time scale how to use machine learning as a tool for designing gestures.

Adjusting model parameters also requires expertise. When presenting systems as interactive installations, the training process was hidden from end users for simplicity. However, as designers, we carefully adjusted the parameters of the models so that the variability of novice demonstrators' gestures would not limit the quality of the sound feedback. For example, the vocalization system used HMR with 20 states and a relatively large regularization which, combined, ensure that the temporal structure of the sound will remain consistent even when the input is noisy. A large regularization means that the prediction will rely more heavily on the time structure and will tolerate larger variations of the input gesture. On the contrary, when designing for expert musical gestures, using lower variance and more states can allow for more fine-grained control. Understanding the role of the model parameters is essential to gain expertise in interaction design with machine learning. To support novice users in this process, we started investigating how visualizations can support the choice of parameters [31]. We proposed to dynamically visualize how changes of the training data and parameters affect the model itself. In future work, we will investigate if and how such visualizations can help designers build a better understanding of the model's underlying mechanisms.

8.3 Designing *through* User Experience: An Iterative Process

We presented an audio application that involves novice users in designing their own gestures for sound control. Our vocalization system was presented as a public installation, and tested by several hundred participants. Our observations of the participants' engagement underline a large diversity of personal strategies for creating gestures and the associated sound. By letting people design by demonstration, our approach allows for experience-driven design of movement interactions. Users can rely on their existing sensori-motor knowledge and past experiences to create personal metaphors of interaction.

As highlighted in the user study, this process is highly dynamic and iterative. Users gradually refine their demonstrations according to the feedback received with direct interaction. Our framework evolved from the initial notion of Mapping-by-Demonstration, that did not describe fully the processes at play when designing with motion-sound mapping with interactive machine learning. Indeed, a focus on demonstrations themselves assumes that the human operators is able to provide high-quality examples that represent a source of 'truth'. Designing efficient gesture sets is a difficult task that requires users to iterate in demonstrating examples and interacting with the trained mapping, thus our focus on the *interactive* experience as the central piece of the design process itself.

9 CONCLUSION

We described an approach to user-centered design of auditory feedback using machine learning. *Mapping through Interaction* relies on an iterative design process in which users can rapidly iterate over (1) recording demonstrations of their personal associations between gestures and sounds and (2) evaluating the mapping learned from these demonstration by directly interacting with the trained system. Our approach relies on probabilistic models of the mapping between sequences of motion and sound parameters, that can be learned from a small set of user-defined examples. We proposed to use Gaussian Mixture Regression and Hidden Markov Regression for real-time joint recognition and mapping.

We presented a concrete application of this approach where users can create personalized auditory feedback by co-producing gestures and vocalizations. An evaluation of the system on a dataset of user-defined gestures and vocalizations showed that Hidden Markov Regression outperforms Gaussian Mixture regression as well as standard implementations of other regression methods. Moreover, our results show that participants rapidly adapt the way they execute the gesture and become increasingly consistent over time. This supports the idea that human learning is essential when using machine learning as a tool for user-centered design: users need to learn how to execute gestures in order to provide high quality demonstrations

ACKNOWLEDGMENTS

We acknowledge all the members of the *{Sound Music Movement} Interaction* team at Ircam. We thank our collaborators for the Vocalization project, in particular Norbert Schnell and Riccardo Borghesi, and for the SoundGuides project: Olivier Chapuis and Sylvain Hanneton. This work was supported by the EDITE school for doctoral studies at Université Pierre et Marie Curie, by the LEGOS project (ANR Grant 31639884), by the Rapid-Mix EU project (H2020-ICT-2014-1 Project ID 644862), by the Labex SMART (ANR-11-LABX-65).

REFERENCES

[1] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 337–346. https://doi.org/10.1145/2702123.2702509

[2] Michael L. Anderson. 2003. Embodied Cognition: A field guide. *Artificial Intelligence* 149, 1 (2003), 91–130. https://doi.org/10.1016/S0004-3702(03)00054-7

[3] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57, 5 (may 2009), 469–483. https://doi.org/10.1016/j.robot.2008.10.024

[4] Gregory Beller. 2014. The Synekine Project. In *Proceedings of the International Workshop on Movement and Computing (MOCO'14)*. ACM, Paris, France, 66–69. https://doi.org/10.1145/2617995.2618007

[5] Penelope A. Best, F. Levy, J. P. Fried, and Fern Leventhal. 1998. Dance and Other Expressive Art Therapies: When Words Are Not Enough. *Dance Research: The Journal of the Society for Dance Research* 16, 1 (jan 1998), 87. https://doi.org/10.2307/1290932

[6] Frédéric Bettens and Todor Todoroff. 2009. Real-time dtw-based gesture recognition external object for max/msp and puredata. *Proceedings of the SMC 2009 Conference* 9, July (2009), 30–35.

[7] Frédéric Bevilacqua, Eric O. Boyer, Jules Françoise, Olivier Houix, Patrick Susini, Agnès Roby-Brami, and Sylvain Hanneton. 2016. Sensori-Motor Learning with Movement Sonification: Perspectives from Recent Interdisciplinary Studies. *Frontiers in Neuroscience* 10 (aug 2016), 385. https://doi.org/10.3389/fnins.2016.00385

[8] Frédéric Bevilacqua, Norbert Schnell, Nicolas Rasamimanana, Bruno Zamborlin, and Fabrice Guédy. 2011. Online Gesture Analysis and Control of Audio Processing. In *Musical Robots and Interactive Multimodal Systems*. Springer, 127–142.

[9] Frédéric Bevilacqua, Bruno Zamborlin, Anthony Sypniewski, Norbert Schnell, Fabrice Guédy, and Nicolas Rasamimanana. 2010. Continuous realtime gesture following and recognition. *Gesture in Embodied Communication and Human-Computer Interaction* (2010), 73–84.

[10] Jeffrey A. Bilmes. 1998. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*. Technical Report.

[11] Matthew Brand and Aaron Hertzmann. 2000. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00)*. ACM Press, New York, New York, USA, 183–192. https://doi.org/10.1145/344779.344865

[12] Bernd Bruegge, Christoph Teschner, Peter Lachenmaier, Eva Fenzl, Dominik Schmidt, and Simon Bierbaum. 2007. Pinocchio. In *Proceedings of the international conference on Advances in computer entertainment technology (ACE '07)*. ACM Press, Salzburg, Austria, 294. https://doi.org/10.1145/1255047.1255132

[13] Sylvain Calinon. 2007. *Continuous extraction of task constraints in a robot programming by demonstration framework.* PhD Dissertation. École Polytechnique Fédéral de Lausanne.

[14] Sylvain Calinon, F. D'halluin, E.L. Sauser, D.G. Caldwell, and Aude Billard. 2010. Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression. *Robotics & Automation Magazine, IEEE* 17, 2 (2010), 44–54.

[15] Sylvain Calinon, Florent Guenter, and Aude Billard. 2007. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 37, 2 (apr 2007), 286–298. https://doi.org/10.1109/TSMCB.2006.886952

[16] Baptiste Caramiaux, Frédéric Bevilacqua, Tommaso Bianco, Norbert Schnell, Olivier Houix, and Patrick Susini. 2014. The Role of Sound Source Perception in Gestural Sound Description. *ACM Transactions on Applied Perception* 11, 1 (apr 2014), 1–19. https://doi.org/10.1145/2536811

[17] Baptiste Caramiaux, Nicola Montecchio, Atau Tanaka, and Frédéric Bevilacqua. 2014. Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 4 (2014), 18:1—-18:34. https://doi.org/10.1145/2643204

[18] Baptiste Caramiaux, Norbert Schnell, Jules Françoise, Frédéric Bevilacqua, Norbert Schnell, and Frédéric Bevilacqua. 2014. Mapping Through Listening. *Computer Music Journal* 38, 34-48 (2014), 34–48. https://doi.org/10.1162/COMJ_a_00255

[19] Baptiste Caramiaux and Atau Tanaka. 2013. Machine Learning of Musical Gestures. In *proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2013)*. Seoul, South Korea.

[20] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, San Jose, CA, USA, 3403–3414. https://doi.org/10.1145/2858036.2858589

[21] Tsuhan Chen. 2001. Audiovisual speech processing. *Signal Processing Magazine, IEEE* 18, 1 (jan 2001), 9–21. https://doi.org/10.1109/79.911195

[22] Kyoungho Choi, Ying Luo, and Jenq-neng Hwang. 2001. Hidden Markov model inversion for audio-to-visual conversion in an MPEG-4 facial animation system. *The Journal of VLSI Signal Processing* 29, 1 (2001), 51–61.

[23] Paul Dourish. 2004. *Where the action is: the foundations of embodied interaction.* The MIT Press.

[24] Jerry Alan Fails and Dan R Olsen. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces (IUI'03)*. 39–45. https://doi.org/10.1145/604045.604056

[25] Sidney Fels and Geoffrey Hinton. 1993. Glove-talkII: A neural network interface between a data-glove and a speech synthesizer. *Neural Networks, IEEE Transactions on* 4, 1 (1993), 2–8.

[26] Rebecca Fiebrink. 2011. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance.* Ph.D. Dissertation. Faculty of Princeton University.

[27] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. 2009. Play-along mapping of musical controllers. In *In Proceedings of the International Computer Music Conference.*

[28] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, Vancouver, BC, Canada, 147. https://doi.org/10.1145/1978942.1978965

[29] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: interactive concept learning in image search. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems (CHI '08)*. 29. https://doi.org/10.1145/1357054.1357061

[30] Jules Françoise. 2015. *Motion-Sound Mapping by Demonstration.* PhD Dissertation. Université Pierre et Marie Curie. http://julesfrancoise.com/phdthesis

[31] Jules Françoise, Frédéric Bevilacqua, and Thecla Schiphorst. 2016. GaussBox: Prototyping Movement Interaction with Interactive Visualizations of Machine Learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, San Jose, CA, 3667–3670. https://doi.org/10.1145/2851581.2890257

[32] Jules Françoise, Baptiste Caramiaux, and Frédéric Bevilacqua. 2012. A Hierarchical Approach for the Design of Gesture-to-Sound Mappings. In *Proceedings of the 9th Sound and Music Computing Conference*. Copenhagen, Denmark, 233–240.

[33] Jules Françoise, Sarah Fdili Alaoui, Thecla Schiphorst, and Frédéric Bevilacqua. 2014. Vocalizing Dance Movement for Interactive Sonification of Laban Effort Factors. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, Vancouver, Canada, 1079–1082. https://doi.org/10.1145/2598510.2598582

[34] Jules Françoise, Norbert Schnell, and Frédéric Bevilacqua. 2013. A Multimodal Probabilistic Model for Gesture-based Control of Sound Synthesis. In *Proceedings of the 21st ACM international conference on Multimedia (MM'13)*. Barcelona, Spain, 705–708. https://doi.org/10.1145/2502081.2502184

[35] Jules Françoise, Norbert Schnell, and Frédéric Bevilacqua. 2014. MaD: Mapping by Demonstration for Continuous Sonification. In *ACM SIGGRAPH 2014 Emerging Technologies (SIGGRAPH '14)*. ACM, Vancouver, Canada, 16. https://doi.org/10.1145/2614066.2614099

[36] Jules Françoise, Norbert Schnell, Riccardo Borghesi, and Frédéric Bevilacqua. 2015. MaD. *interactions* 22, 3 (2015), 14–15. https://doi.org/10.1145/2754894

[37] Karmen Franinović and Stefania Serafin. 2013. *Sonic Interaction Design.* MIT Press.

[38] Shengli Fu, Ricardo Gutierrez-Osuna, Anna Esposito, Praveen K. Kakumanu, and Oscar N. Garcia. 2005. Audio/visual mapping with cross-modal hidden Markov models. *Multimedia, IEEE Transactions on* 7, 2 (apr 2005), 243–252. https://doi.org/10.1109/TMM.2005.843341

[39] Zoubin Ghahramani and Michael I. Jordan. 1994. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems.*

[40] Nicholas Gillian and Joseph A Paradiso. 2014. The Gesture Recognition Toolkit. *Journal of Machine Learning Research* 15 (2014), 3483–3487. http://jmlr.org/papers/v15/gillian14a.html

[41] Nicholas Edward Gillian. 2011. *Gesture Recognition for Musician Computer Interaction.* PhD dissertation. Faculty of Arts, Humanities and Social Sciences.

[42] Marco Gillies, Harry Brenton, and Andrea Kleinsmith. 2015. Embodied Design of Full Bodied Interaction with Virtual Humans. In *Proceedings of the 2Nd International Workshop on Movement and Computing (MOCO '15)*. ACM, Vancouver, British Columbia, Canada, 1–8. https://doi.org/10.1145/2790994.2790996

[43] Rolf Inge Godøy, Egil Haga, and A.R. Jensenius. 2006. Exploring music-related gestures by sound-tracing - a preliminary study. In *2nd ConGAS International Symposium on Gesture Interfaces for Multimedia Systems*. 9–10.

[44] Rolf Inge Godøy, Egil Haga, and A. Jensenius. 2006. Playing âĂIJ Air Instruments âĂİ: Mimicry of Sound-producing Gestures by Novices and Experts. *Gesture in Human-Computer Interaction and Simulation* (2006), 256–267.

[45] Vincent Goudard, Hugues Genevois, Émilien Ghomi, and Boris Doval. 2011. Dynamic Intermediate Models for audiographic synthesis. In *Proceedings of the Sound and Music Computing Conference (SMC'11)*.

[46] Camille Goudeseune. 2002. Interpolated mappings for musical instruments. *Organised Sound* 7, 2 (2002), 85–96.

[47] Thomas Hermann, John G. Neuhoff, and Andy Hunt. 2011. *The Sonification Handbook.* Logos Verlag, Berlin, Germany.

[48] Kristina Höök, Martin P Jonsson, Anna Ståhl, and Johanna Mercurio. 2016. Somaesthetic Appreciation Design. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, San Jose, CA, USA, 3131–3142. https://doi.org/10.1145/2858036.2858583

[49] Thomas Hueber and Pierre Badin. 2011. Statistical Mapping between Articulatory and Acoustic Data, Application to Silent Speech Interface and Visual Articulatory Feedback. *Proceedings of the 1st International Workshop on Performative Speech and Singing Synthesis (p3s)* (2011).

[50] Andy Hunt, Marcelo M. Wanderley, and Ross Kirk. 2000. Towards a Model for Instrumental Mapping in Expert Musical Interaction. In *Proceedings of the 2000 International Computer Music Conference*. 209–212.

[51] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation* 25, 2 (feb 2013), 328–73. https://doi.org/10.1162/NECO_a_00393

[52] Andrew Johnston. 2009. *Interfaces for musical expression based on simulated physical models.* Ph.D. Dissertation. University of Technology, Sydney.

[53] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2004. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 3 (may 2004), 358–386. https://doi.org/10.1007/s10115-004-0154-9

[54] David Kirsh. 2013. Embodied cognition and the magical future of interaction design. *ACM Transactions on Computer-Human Interaction* 20, 111 (2013), 3:1–3:30. https://doi.org/10.1145/2442106.2442109

[55] David Kirsh, Dafne Muntanyola, and RJ Jao. 2009. Choreographic methods for creating novel, high quality dance. In *5th International workshop on Design and Semantics of Form and Movement.*

[56] Andrea Kleinsmith and Marco Gillies. 2013. Customizing by doing for responsive video game characters. *International Journal of Human Computer Studies* 71, 7-8 (2013), 775–784. https://doi.org/10.1016/j.ijhcs.2013.03.005

[57] Paul Kolesnik and Marcelo M. Wanderley. 2005. Implementation of the Discrete Hidden Markov Model in Max/MSP Environment. In *FLAIRS Conference*. 68–73.

[58] Todd Kulesza, Margaret Burnett, Weng-keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15*. Atlanta, GA, USA., 126–137. https://doi.org/10.1145/2678025.2701399

[59] Michael Lee, Adrian Freed, and David Wessel. 1992. Neural networks for simultaneous classification and parameter estimation in musical instrument control. *Adaptive and Learning Systems* 1706 (1992), 244–55.

[60] Marc Leman. 2008. *Embodied Music Cognition and mediation technology.* The MIT Press.

[61] Moira Logan. 1984. Dance in the schools: A personal account. *Theory Into Practice* 23, 4 (sep 1984), 300–302. https://doi.org/10.1080/00405848409543130

[62] Elena Márquez Segura, Laia Turmo Vidal, Asreen Rostami, and Annika Waern. 2016. Embodied Sketching. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, San Jose, CA, USA, 6014–6027. https://doi.org/10.1145/2858036.2858486

[63] Eduardo R. Miranda and Marcelo M. Wanderley. 2006. *New digital musical instruments: control and interaction beyond the keyboard.* AR Editions, Inc.

[64] Paul Modler. 2000. Neural Networks for Mapping Hand Gestures to Sound Synthesis parameters. *Trends in Gestural Control of Music* (2000), 301–314.

[65] Ali Momeni and Cyrille Henry. 2006. Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis. *Computer Music Journal* 30, 1 (2006), 49–66.

[66] Kevin Patrick Murphy. 2012. *Machine Learning: A Probabilistic Perspective.* MIT press, Cambridge, Massachusetts.

[67] Uran Oh and Leah Findlater. 2013. The challenges and potential of end-user gesture customization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, Paris, France, 1129. https://doi.org/10.1145/2470654.2466145

[68] Dirk Ormoneit and Volker Tresp. 1996. Improved Gaussian Mixture Density Estimates Using Bayesian Penalty Terms and Network Averaging. In *Advances in Neural Information Processing Systems 8*, D S Touretzky, M C Mozer, and M E Hasselmo (Eds.). MIT Press, 542–548.

[69] Kayur Patel, Naomi Bancroft, Steven M Drucker, James Fogarty, Andrew J Ko, and James Landay. 2010. Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 37–46. https://doi.org/10.1145/1866029.1866038

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[71] Lawrence R. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[72] Nicolas Rasamimanana, Frédéric Bevilacqua, Norbert Schnell, Emmanuel Fléty, and Bruno Zamborlin. 2011. Modular Musical Objects Towards Embodied Control Of Digital Music Real Time Musical Interactions. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI'11).* Funchal, Portugal, 9–12.

[73] Davide Rocchesso, Guillaume Lemaitre, Patrick Susini, Sten Ternström, and Patrick Boussard. 2015. Sketching Sound with Voice and Gesture. *interactions* 22, 1 (2015), 38–41. https://doi.org/10.1145/2685501

[74] Joseph B. Rovan, Marcelo M. Wanderley, Shlomo Dubnov, and Philippe Depalle. 1997. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *Proceedings of the AIMI International Workshop.* 68–73.

[75] Stefan Schaal. 1999. Is imitation learning the route to humanoid robots. *Trends in cognitive sciences* 3, 6 (1999), 233–242.

[76] Stefan Schaal, Auke Ijspeert, and Aude Billard. 2003. Computational approaches to motor learning by imitation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 358, 1431 (mar 2003), 537–47. https://doi.org/10.1098/rstb.2002.1258

[77] Thecla Schiphorst. 2009. soft(n). In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI EA '09).* ACM, Boston, MA, USA, 2427. https://doi.org/10.1145/1520340.1520345

[78] Diemo Schwarz. 2007. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine* 24, 2 (2007), 92–104.

[79] Hsi Guang Sung. 2004. *Gaussian Mixture Regression and Classification.* PhD Dissertation. Rice University, Houston, TX.

[80] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09).* Boston, USA, 1283–1292. https://doi.org/10.1145/1518701.1518895

[81] Joëlle Tilmanne. 2013. *Data-driven Stylistic Humanlike Walk Synthesis.* PhD Dissertation. University of Mons.

[82] Joëlle Tilmanne, Alexis Moinet, and Thierry Dutoit. 2012. Stylistic gait synthesis based on hidden Markov models. *EURASIP Journal on Advances in Signal Processing* 2012, 1 (2012), 72. https://doi.org/10.1186/1687-6180-2012-72

[83] Tomoki Toda, Alan W. Black, and Keiichi Tokuda. 2004. Acoustic-to-articulatory inversion mapping with gaussian mixture model.. In *INTERSPEECH.*

[84] Tomoki Toda, Alan W Black, and Keiichi Tokuda. 2008. Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. *Speech Communication* 50, 3 (2008), 215–227.

[85] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. 2013. Speech Synthesis Based on Hidden Markov Models. *Proc. IEEE* 101, 5 (2013), 1234–1252. https://doi.org/10.1109/JPROC.2013.2251852

[86] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 2000. Speech parameter generation algorithms for HMM-based speech synthesis. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, Vol. 3. 1315–1318. https://doi.org/10.1109/ICASSP.2000.861820

[87] Doug Van Nort, Marcelo M. Wanderley, and Philippe Depalle. 2004. On the choice of mappings based on geometric properties. In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME'04).* National University of Singapore, 87–91.

[88] Doug Van Nort, Marcelo M Wanderley, and Philippe Depalle. 2014. Mapping Control Structures for Sound Synthesis: Functional and Topological Perspectives. *Comput. Music J.* 38, 3 (2014), 6–22. https://doi.org/10.1162/COMJ_a_00253

[89] Freya Vass-Rhee. 2010. Dancing music: The intermodality of The Forsythe Company. In *William Forsythe and the Practice of Choreography*, Steven Spier (Ed.). 73–89.

[90] Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. 2009. User-defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09).* ACM, 1083–1092. https://doi.org/10.1145/1518701.1518866

[91] Bruno Zamborlin. 2015. *Studies on customisation-driven digital music instruments.* PhD Dissertation. Goldsmith University of London and Université Pierre et Marie Curie.

[92] Heiga Zen, Yoshihiko Nankaku, and Keiichi Tokuda. 2011. Continuous Stochastic Feature Mapping Based on Trajectory HMMs. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 2 (feb 2011), 417–430. https://doi.org/10.1109/TASL.2010.2049685

[93] Heiga Zen, Keiichi Tokuda, and Tadashi Kitamura. 2007. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech & Language* 21, 1 (jan 2007), 153–173. https://doi.org/10.1016/j.csl.2006.01.002

[94] Le Zhang and Steve Renals. 2008. Acoustic-Articulatory Modeling With the Trajectory HMM. *IEEE Signal Processing Letters* 15 (2008), 245–248. https://doi.org/10.1109/LSP.2008.917004

Received December 2016; revised June 2017; revised February 2018; accepted April 2018