Motion-sound Mapping By Demonstration

Jules Françoise

▶ **To cite this version:**

Jules Françoise. Motion-sound Mapping By Demonstration. Other [cs.OH]. Université Pierre et Marie Curie - Paris VI, 2015. English. NNT : 2015PA066105 . tel-01206009

HAL Id: tel-01206009
https://theses.hal.science/tel-01206009

Submitted on 28 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MOTION-SOUND MAPPING BY DEMONSTRATION

Jules FRANÇOISE

UNIVERSITÉ PIERRE ET MARIE CURIE

École doctorale Informatique, Télécommunications et Électronique

Institut de Recherche et Coordination Acoustique/Musique

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité **Informatique**

soutenue le 18 mars 2015 devant le jury composé de :

Catherine ACHARD	Examinatrice	Université Pierre et Marie Curie
Thierry ARTIÈRES	Directeur de thèse	Université Pierre et Marie Curie
Frédéric BEVILACQUA	Directeur de thèse	Ircam
Olivier CHAPUIS	Examinateur	Université Paris-Sud
Thierry DUTOIT	Rapporteur	Université de Mons
Rebecca FIEBRINK	Examinatrice	Goldsmith University of London
Sergi JORDÀ	Examinateur	Universitat Pompeu Fabra
Marcelo WANDERLEY	Rapporteur	McGill University

Jules Françoise: *Motion-Sound Mapping By Demonstration. Final Version* as of June 9, 2015.

SUPERVISORS:
Frédéric Bevilacqua
Thierry Artières

Abstract

Designing the relationship between motion and sound is essential to the creation of interactive systems. This thesis proposes an approach to the design of the mapping between motion and sound called Mapping-by-Demonstration. Mapping-by-Demonstration is a framework for crafting sonic interactions from demonstrations of embodied associations between motion and sound. It draws upon existing literature emphasizing the importance of bodily experience in sound perception and cognition. It uses an interactive machine learning approach to build the mapping iteratively from user demonstrations.

Drawing upon related work in the fields of animation, speech processing and robotics, we propose to fully exploit the generative nature of probabilistic models, from continuous gesture recognition to continuous sound parameter generation. We studied several probabilistic models under the light of continuous interaction. We examined both instantaneous (Gaussian Mixture Model) and temporal models (Hidden Markov Model) for recognition, regression and parameter generation. We adopted an Interactive Machine Learning perspective with a focus on learning sequence models from few examples, and continuously performing recognition and mapping. The models either focus on movement, or integrate a joint representation of motion and sound. In movement models, the system learns the association between the input movement and an output modality that might be gesture labels or movement characteristics. In motion-sound models, we model motion and sound jointly, and the learned mapping directly generates sound parameters from input movements.

We explored a set of applications and experiments relating to real-world problems in movement practice, sonic interaction design, and music. We proposed two approaches to movement analysis based on Hidden Markov Model and Hidden Markov Regression, respectively. We showed, through a use-case in Tai Chi performance, how the models help characterizing movement sequences across trials and performers. We presented two generic systems for movement sonification. The first system allows users to craft hand gesture control strategies for the exploration of sound textures, based on Gaussian Mixture Regression. The second system exploits the temporal modeling of Hidden Markov Regression for associating vocalizations to continuous gestures. Both systems gave birth to interactive installations that we presented to a wide public, and we started investigating their interest to support gesture learning.

Résumé

Le design du mapping (ou couplage) entre mouvement et son est essentiel à la création de systèmes interactifs sonores et musicaux. Cette thèse propose une approche appelée mapping par démonstration qui permet aux utilisateurs de créer des interactions entre mouvement et son par des exemples de gestes effectués pendant l'écoute. L'approche s'appuie sur des études existantes en perception et cognition sonore, et vise à intégrer de manière plus cohérente la boucle action-perception dans le design d'interaction. Le mapping par démonstration est un cadre conceptuel et technique pour la création d'interactions sonores à partir de démonstrations d'associations entre mouvement et son. L'approche utilise l'apprentissage automatique interactif pour construire le mapping à partir de démonstrations de l'utilisateur.

En s'appuyant sur des travaux récents en animation, en traitement de la parole et en robotique, nous nous proposons d'exploiter la nature générative des modèles probabilistes, de la reconnaissance de geste continue à la génération de paramètres sonores. Nous avons étudié plusieurs modèles probabilistes, à la fois des modèles instantanés (Modèles de Mélanges Gaussiens) et temporels (Modèles de Markov Cachés) pour la reconnaissance, la régression, et la génération de paramètres sonores. Nous avons adopté une perspective d'apprentissage automatique interactif, avec un intérêt particulier pour l'apprentissage à partir d'un nombre restreint d'exemples et l'inférence en temps réel. Les modèles représentent soit uniquement le mouvement, soit intègrent une représentation conjointe des processus gestuels et sonores, et permettent alors de générer les trajectoires de paramètres sonores continûment depuis le mouvement.

Nous avons exploré un ensemble d'applications en pratique du mouvement et danse, en design d'interaction sonore, et en musique. Nous proposons deux approches pour l'analyse du mouvement, basées respectivement sur les modèles de Markov cachés et sur la régression par modèles de Markov. Nous montrons, au travers d'un cas d'étude en Tai Chi, que les modèles permettent de caractériser des séquences de mouvements entre plusieurs performances et différents participants. Nous avons développé deux systèmes génériques pour la sonification du mouvement. Le premier système permet à des utilisateurs novices de personnaliser des stratégies de contrôle gestuel de textures sonores, et se base sur la régression par mélange de Gaussiennes. Le second système permet d'associer des vocalisations à des mouvements continus. Les deux systèmes ont donné lieu à des installations publiques, et nous avons commencé à étudier leur application à la sonification du mouvement pour supporter l'apprentissage moteur.

I'm not going to write about love. I'm going to
write only about the weather.
The weather in Berlin is nice today.

— Viktor Shklovsky.

Acknowledgments

My biggest acknowledgment goes to my supervisor Frédéric Bevilacqua, even though no word would fully express my gratitude, admiration and pleasure. Fred is a rare man, and my thesis would not have been the same (or at all) without him. He has been present at all times, supporting my work but also my uncertainties and my joys. These three years with him have been incredibly enriching, and in the end I don't feel indebted but grateful and deeply happy for all the time we shared. For these many 'discussions', that consistently ended with both of us gazing in the vague, with enough ideas to think for a long time. For the time at work and beyond. I have no doubt this time will sustain.

I also wish to thank Thierry Artières, who made this thesis possible. I am infinitely happy that you gave me this opportunity, and I am grateful for the support and the rich discussions on the most technical aspects of my work.

Baptiste, who first welcomed me in the team when I was a master's student: the time we shared at work was unforgettable, as much as the friendship that followed. You keep inspiring me and I hope this friendship continues.

Now starts the long and incomplete list of people that contributed to my work, my personal life, that supported me one way or another along this chapter of my life. First of all, I acknowledge all the personal from Ircam for giving me a rich, motivating, and dynamic environment to work: Hugues, Sylvie, Carole, Martine, and so many others. My colleagues from the ISMM team, with whom I shared so many fruitful discussions and ideas, dinners, drinks and late nights. Norbert, for the constant support and the numerous contributions to my work, for your theories and inspiring ideas. Diemo, for the numerous advices and musical influences; Riccardo, who as always been present and supportive; Sarah, for the constant fun and intense collaboration, who gave me the opportunity to come to Vancouver! Eric, Jean-Philippe, Kevin, Tommaso, Fabien, that shared my office downstairs and the JS team upstairs: Victor, Robi, Benjamin, Karim, ... I also wish to thank Pablo, for the great work and friendly collaboration.

I had the chance to work with a number of people, that greatly inspired me through collaborations and informal discussions. Among others, I wish to thank Greg Beller, Ianis Lallemand, Thecla Schiphorst, Sylvain Hanneton, Agnès Robi-Bramy, Olivier Chapuis and Olivier Houix.

I would like to thank my family: my parents, Alice and Paul. Your constant support has been essential, and this thesis would not have been possible without you.

Nothing would have been the same without all my friends, the whole *Zoophoniq* team, my roommates, and so many others: Hélène, who will always have a special place, Thomas, Yves, Cécile, Erwan, Brett, Isaline, Mathieu, Ambroise, Myrtille, Marine, Ana, Sandra, Marc, Jean-Marie, Chloé, Jérôme, José, to name a few...

Of course, Маша, and I guess you already know it.

Contents

List of Figures

List of Equations

Glossary

ANOVA	Analysis Of Variance.
ASR	Automatic Speech Recognition.
CI	confidence interval.
DBN	Dynamic Bayesian Network.
DMI	Digital Musical Instrument.
DTW	Dynamic Time Warping.
EM	Expectation-Maximization.
GMM	Gaussian Mixture Model.
GMR	Gaussian Mixture Regression.
HCI	Human-Computer Interaction.
HHMM	Hierarchical Hidden Markov Model.
HMM	Hidden Markov Model.
HMR	Hidden Markov Regression.
HSMM	Hidden Semi-Markov Model.
IML	Interactive Machine Learning.
k-NN	k-Nearest-Neighbor.
LSE	Least Squares Estimate.
MAP	Maximum A Posteriori.
MbD	Mapping-by-Demonstration.
MFCC	Mel-Frequency Cepstral Coefficient.
ML	Maximum Likelihood.
MLLR	Maximum Likelihood Linear Regression.
MLPG	Maximum Likelihood Parameter Generation.
NIME	New Interfaces for Musical Expression.
PbD	Programming-by-Demonstration.
pdf	probability density function.
RMS	Root Mean Square.
SID	Sonic Interaction Design.
TTS	Text-To-Speech.

On more poetic occasions a musician will speak as if the instrument has come to know something of its player. It would seem quite natural then to think about intelligent instruments that could adapt in some automated way to a personal playing style.

David Wessel (Wessel, 1991)

1 Introduction

1.1 Background and General Aim

We often experience sound through movement. As we move along music or produce sound through actions, our movements shape the way we perceive sound. While action and perception are now considered intrinsically linked by neuroscientists, few approaches tightly incorporate perception and experience in sound design practice and tools. This thesis aims to bridge a gap between experience and design through the development of a Mapping-by-Demonstration (MbD) approach that let users craft interactive systems through examples of movement and sound; therefore supporting a shift from designing *for* user experience to designing *through* user experience.

Technological, contextual, artistic, or social issues determine the choices of sound designers, artists, or technologists in the design of digital artifacts. These factors critically impact the nature and properties of the resulting systems that vary in accuracy, expressiveness, ease of use, etc. Users might often want to express their idiosyncrasies, especially as expertise increases, which leads to blur the lines between the designer and the user. Therefore, designs and tools for design must be as versatile as their usage presents variations, and might often be adapted to context-, application-, and user-specific needs. Such requirements relate to the challenges of user customization, context and ecological validity identified by LaViola (2013) in a recent review of three-dimensional gestural interaction.

While several mapping design methods are based on establishing direct links between parameters, other recent approaches rely on intermediate models of interaction. In particular, machine learning is gaining interest as a tool for data-driven design of the mapping. Interactive Machine Learning (IML) emphasizes the role of the user as the central actor in making machine learning efficient and expressive. The user therefore actively contributes to the learning process by providing input data, training models and evaluating results in a short interaction loop (Fiebrink, 2011). These developments in the music community echo the *Programming-by-Demonstration* methodology in robotics (Billard et al., 2008), that empha-

sizes the role of the human in the specification of desirable behaviors through embodied demonstrations.

In this work, we investigate Interactive Machine Learning as a way to integrate the action-perception loop as a fundamental aspect of mapping design. We consider learning the mapping from demonstrations of embodied associations between motion and sound.

1.2 Probabilistic Models: from recognition to synthesis

Interactive Machine Learning focuses on user-centered approaches to the design of learning methods. Its application to designing interactions between motion and sound leads to several requirements, that motivate the choice of a probabilistic approach.

In multimodal interactive systems, uncertainty arises at a number of levels, from movement execution and measurement noise to recognition and generation. Uncertainty is therefore ubiquitous in real-world applications, where most observations and predictions cannot be made with complete certainty. Many approaches to machine learning rely on a probabilistic approach to handle uncertainty.[1] In particular, probabilistic graphical models provide a consistent framework for updating beliefs by combining prior knowledge with new evidence (Koller and Friedman, 2009).

Probabilistic sequence models such as Hidden Markov Models (HMMs) have a long history in gesture recognition and analysis for their ability to handle both timing and dynamic variations. Bevilacqua et al. (2010) proposed a template-based model that learns from a single example and allows for continuous recognition and alignment. Caramiaux (2014) further extended continuous movement analysis to the characterization of gesture variations. Along this work, we aim to address how movement models can efficiently implement online continuous movement analysis. Besides, we aim to formalize the design strategies based on recognition, to support this shift from a *classification-triggering* discrete interaction paradigm to continuous interaction.

Considering the pioneering field of speech processing, which inspired many developments in gesture recognition and movement analysis, it is interesting to note how the use of generative sequence models progressively transitioned from recognition to synthesis. As reviewed by Tokuda et al. (2013), HMM-based speech synthesis gained interest in recent years as a parametric synthesis method that provides flexible control over the generated parameter sequences. It allows, for example, to modify speaker characteristics (mimicking, mixing, producing voices), or to produce expressive speech by integrating articulatory and affective features. While the method proved to be efficient for expressive movement synthesis (Tilmanne et al., 2012), its most interesting applications concern the modeling of cross-modal relationships. Several domains investigate movement generation from speech (speech-driven character animation (Fu et al., 2005), acoustic-articulatory

1 While most of the methods considered in this dissertation relate to Bayesian inference, we prefer to adopt the term 'probabilistic' approach, as discussed by (Murphy, 2012, Preface)

inversion (Zhang and Renals, 2008)), speaker conversion (Zen et al., 2011), or speech synthesis driven by articulatory movements (silent speech interfaces (Hueber and Badin, 2011)). Such applications aim to learn a complex cross-modal relationship, which relates to our interests in motion-sound mapping. Moreover, they emphasize continuity and variations (e.g. in articulation, prosody, affect) as essential to capturing expression.

Our field of study is concerned with the design of the relationships between motion and sound for Digital Musical Instruments (DMIs), and Sonic Interaction Design. Thus, it differs from the above domains of motion-speech modeling in a number of ways. First, we have strong real-time constraints: interactive sonic systems require the sound synthesis to be instantaneously and continuously driven by the movement. This prevents the use of offline inference, that is mostly exploited in the most of the above applications. Second, while speech-related movements and sounds emerge from the same action, we focus on allowing users to define arbitrary motion-sound relationships. The nature and properties of such relationships highly depends on contextual and personal factors.

Throughout this work, we investigate how such models can be used for embodied design of motion-sound relationships. In particular, we examine their ability to learn from few examples and perform predictions in real-time. We study how articulating probabilistic recognition and generation opens novel possibilities for mapping design.

1.3 Mapping by Demonstration: Concept and Contributions

This thesis approaches the design of the relationships between motion and sound through a Mapping-by-Demonstration (MbD) perspective. Our work is concerned with two principal ideas. Our first objective is the development of models that consistently encode the relationships between motion and sound with joint representations. Second, we aim at integrating users in a design process that emphasizes the action-perception loop. This thesis offers a set of contributions that relate to conceptual, theoretical, practical, and experimental issues.

First, we review mapping strategies between motion and sound in the field of music and interactive sonification. From the body of work investigating the use of machine learning in sound and music interaction design, we formalize the problem of Mapping-by-Demonstration (MbD), highlighting multimodality and temporality as key aspects of motion-sound relationships modeling. We emphasize the need for probabilistic models that account for uncertainty in both movement and sound processes. Drawing upon related work in the fields of animation, speech processing and robotics, we propose to fully exploit the generative nature of probabilistic models from continuous gesture recognition to continuous sound parameter generation. We argue that probabilistic models provide a fertile ground for continuous interaction as they allow for real-time, flexible, and parametric control of audio processing through parameter generation.

Second, we study several probabilistic models under the light of continuous interaction. We examine both instantaneous (Gaussian Mixture Model) and temporal models (Hidden Markov Model) for recognition, regression and parameter generation. We adopt an Interactive Machine Learning perspective with a focus on learning sequence models from few examples, and continuously performing recognition and mapping. The models either focus on movement only, or integrate a joint representation of motion and sound. In movement models, we aim at learning the association between the input movement and an output modality that might be gesture labels or movement characteristics. In motion-sound models, we model motion and sound jointly, and the learned mapping directly generates sound parameters from input movements.

Finally, we explore a set of applications and experiments relating to real-world problems in movement practice, sonic interaction design, and music. We present two generic systems addressing interactive control of recorded sounds, and vocalization in movement performance and learning, respectively.

1.4 Outline of the Dissertation

CHAPTER 2 gives an overview of the related work in motion-sound mapping with a machine learning perspective. We propose to integrate the notions of action-perception and we outline related research in other fields such as speech processing and robotics.

CHAPTER 3 formalizes the concept of Mapping-by-Demonstration. We propose a general architecture and detail the main components of the framework, addressing both human factors and technological issues of the methodology.

CHAPTER 4 details probabilistic movement models. Focusing on user-centered learning from few examples, we outline the representation, learning and inference algorithms for Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and Hierarchical Hidden Markov Models (HHMMs). Importantly, we present initial applications of the Mapping-by-Demonstration framework, and we discuss strategies for sonic interaction design based on continuous gesture recognition.

CHAPTER 5 applies the probabilistic movement models introduced in Chapter 4 to movement analysis. We examine various models for gesture segmentation, recognition and spotting, and illustrate the use of how such methods for online performance analysis.

CHAPTER 6 introduces cross-modal probabilistic models for motion-sound mapping. We detail the mechanisms and implementation of Gaussian Mixture Regression and Hidden Markov Regression, and we discuss the advantages of the probabilistic framework for sound parameter generation.

CHAPTER 7 exploits the generative models presented in Chapter 6 for movement analysis across different performers and for cross-modal analysis of vocalized movements.

CHAPTER 8 introduces a generic system for gesture-based control of sound textures. The system uses Gaussian Mixture Regression for mapping between hand movements and sound descriptors. We present an interactive installation and a controlled experiment investigating gesture imitation with sound feedback.

CHAPTER 9 presents a system for associating continuous movements to vocalizations. Based on Hidden Markov Regression, the system learns a mapping from dynamic gestures to vocal sounds. It is applied to a public installation, and to movement sonification in dance pedagogy.

CHAPTER 10 concludes by summarizing the contributions and by discussing the limitations and future research.

APPENDIX A presents our publications in relationship with the present manuscript, and reports on the XMM library that implements the probabilistic models studied in this dissertation.

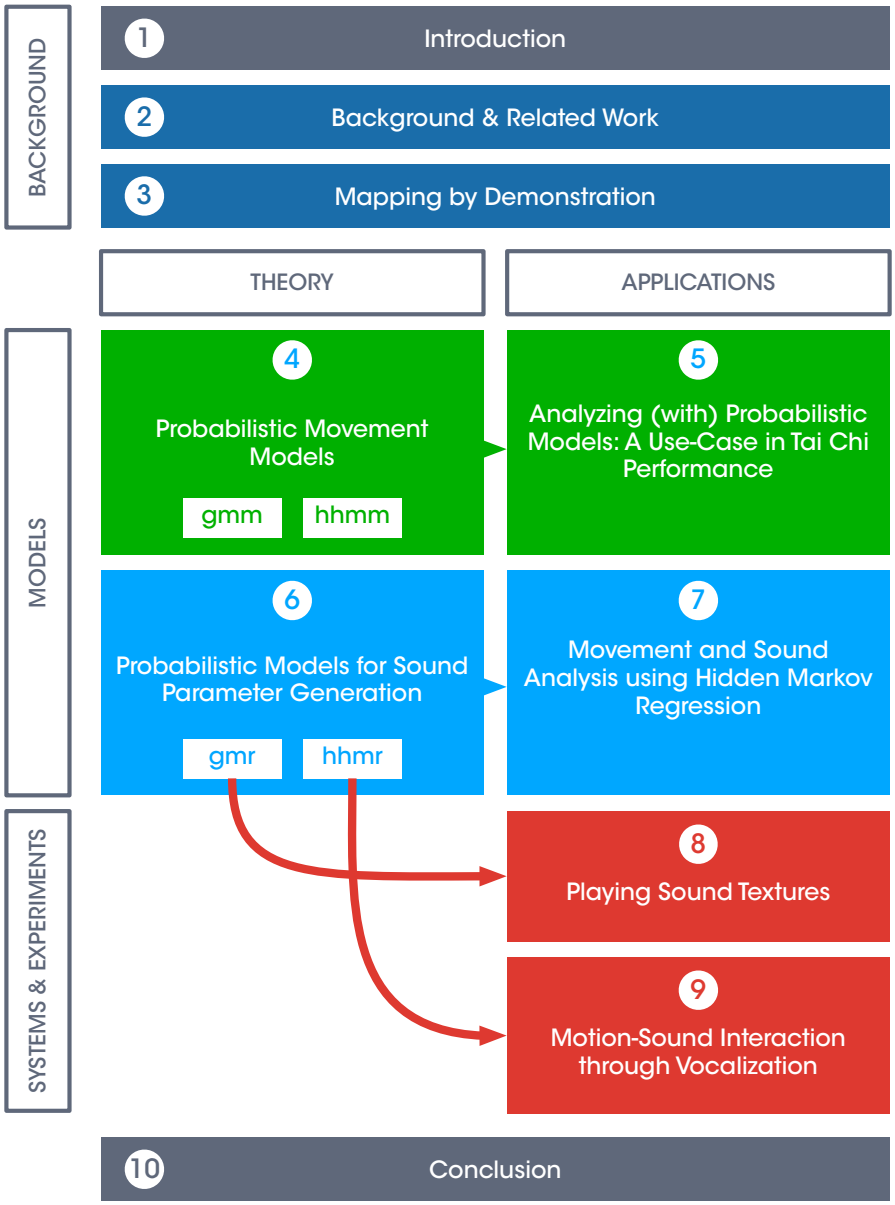


Figure 1.1: Graphical Outline of the Dissertation.

2

Background & Related Work

In this chapter, we motivate the present research with an overview of the related work. We start by reviewing in Section 2.1 the concepts and technological approaches to mapping design in Digital Musical Instruments, with a particular interest in method based on gesture recognition and machine learning (Section 2.2). Then, we review the emerging field of Interactive Machine Learning and its current threads in music computing (Section 2.3). After outlining current research in embodied music cognition, we motivate in Section 2.4 the Mapping-by-Demonstration framework as the intersection between the mapping through listening design principle and interactive machine learning. Finally, we give an overview of the related research in the fields of speech processing and robotics in Section 2.5, that motivates the use of probabilistic modeling.

2.1 Motion-Sound Mapping: from Wires to Models

Our work is in line with the branch of music computing involving the body in interaction with sound and music, with a focus on designing the relationship between motion and sound. It strongly relate to the fields of New Interfaces for Musical Expression (NIME),[1] Digital Musical Instrument (DMI) design (Miranda and Wanderley, 2006), Sonic Interaction Design (SID) (Franinović and Serafin, 2013), and sonification (Hermann et al., 2011). Thereafter, we restrict our survey of the literature to the body of work dealing with the so called mapping between performers' movements and sound synthesis.

Since the early experiments of Max Mathews with the Radio Baton in the 1980s — or "Pitch, the most important and least expressive part of music" (Mathews, 1991), — designing the relationship between gesture and sound has been central to research and practice of interactive digital music systems. This relationship between motion parameters and sound control parameters has been formalized as *mapping* (Rovan et al., 1997). Specific choices in mapping design impact on the interaction possibilities, for ex-

1 NIME website: http://nime.org/

ample in terms of ease-of-use, expressivity, controllability, or metaphorical qualities.

In this section we give an overview of the approaches to the design of the mapping that have been proposed within the New Interfaces for Musical Expression (NIME) community. Some are based on an analytical formulation of the mapping, which is then built by *wiring* gesture parameters to sound parameters. Other strategies take advantage of intermediate models allowing to implement particular behaviors and metaphors. This work is in line with recent developments in the use of machine learning for mapping design.

Note: Mapping has often been defined as the layer connecting motion sensors to sound synthesis parameters (Rovan et al., 1997). This definition is obviously advantageous from a technical point of view, but can be limited to describe the entire interaction model between the human and the sound output. In this section, we review several approaches to mapping design mostly under this initial definition, and we further discuss this terminological issue at the end of this survey (Section 2.4).

2.1.1 Explicit Mapping

Hunt et al. (2000) proposed a terminology for mapping analysis that distinguishes between *explicit* and *implicit* mapping strategies for musical performance. *Explicit* design refers to strategies in which gesture parameters are directly *wired* to sound control parameters. Wanderley and collaborators contributed to analyze explicit mapping in more details through the definition of one-to-one, one-to-many and many-to-one strategies (Rovan et al., 1997; Wanderley, 2001; Hunt and Wanderley, 2002). Hunt and Kirk (2000) show that if simple *one-to-one* mapping strategies are easily understandable, their expressive power in the long term is more limited than complex *one-to-many* or *many-to-many* mapping strategies. A similar analysis can be found in (Verfaille et al., 2006), who extended the question to simultaneous gesture control and adaptive control of digital audio effects.

Yet, the expressive power and consistency of direct relationships can be limited when the sound synthesis parameters are not perceptually meaningful. Rovan et al. (1997) argued that direct mapping strategies are particularly relevant to physical modeling sound synthesis, because the input parameters of the synthesis model already have a physical significance, which can be easily related to gesture parameters. Number of works have studied controllers and mapping strategies for the control of physical models (Serafin et al., 2001; Demoucron, 2008), often stressing the importance of haptic feedback (Howard and Rimell, 2004; Cadoz, 1988; Castagne et al., 2004).

Arfib et al. (2002) proposed to address this problem through a 3-layer mapping for the control of physical modeling sound synthesis, that extends gesture and sound representations with perceptual spaces. Arfib et al. argued that the integration of perceptual descriptors facilitates the creation of mapping strategies with respect to metaphors or navigations paradigms.

> Explicit mapping refers to strategies that directly 'wire' motion parameters to sound parameters. The method is efficient and expressive, but can be limited by the lack of perceptual significance of the sound parameters.

Implicit mapping represents an alternative approach to overcome some of the limitations of explicit strategies. Implicit mapping strategies use an intermediate model to encode a possibly complex behavior at the interface between motion and sound parameters. Such models can take a variety of forms, from interpolation maps to dynamical systems and recognition-based approaches.

2.1.2 Mapping with Physical Models

We mentioned above how physical modeling sound synthesis can address the lack of significance of sound control parameters. For sound synthesis, physical modeling often aims to emulate the behavior of known instruments; or at least, the acoustic behavior of vibrating objects. More flexible approaches have been proposed where physical models serve as an interaction between motion parameters and sound synthesis. Such models are often inspired by dynamical systems modeling, but they aim at creating a novel, autonomous behavior rather than emulating a specific physical structure.

The work of Cadoz et al. at ACROE built the foundations of dynamical systems for sound synthesis and gestural interaction (Cadoz et al., 1993; Castagné and Cadoz, 2002). The PMPD physical modeling software presented in Henry (2004) introduces a collection of low-level components for the design of mass-spring dynamical models in PureData with applications to real-time interaction with audio synthesis. The modularity of the software allows users to create number of topologies to explore various types of interactions, keeping the mapping intuitive because physically plausible. Momeni and Henry (2006) extended this approach to concurrent control of audio and video. They propose dynamic independent mapping layers based on dynamical models to map between perceptual spaces of motion and sound. The mapping exhibits time-variable behaviors, adding complexity in the interaction, without altering the transparency of the relationship between motion and sound. Extending this concept to include physical models, topological models and genetic models, Goudard et al. (2011) introduced the notion of Dynamic Intermediate Model, that defines the layer between hardware interfaces and audio processing as a composition of models with dynamic behaviors.

Johnston gave an HCI perspective on mapping with dynamical models in DMIs with a focus on user-centered evaluation (Johnston et al., 2007; Johnston, 2009). Johnston studied the influence of dynamical models mapping of live sound of acoustic instruments to audio and visuals on the performer's music making. The results show that even simple physical models can create interesting interactions, thanks to an intuitive understanding of the behavior of the interface. Johnston identified three modes of interaction: instrumental, ornamental and conversational interaction; mostly discriminated by the degree of controllability and dialog with the virtual in-

strument (Johnston et al., 2008). Similarly, Pirrò and Eckel (2011) observed that performers quickly reach an intimate control of the instrument, because it corresponds to a known physical behavior.

> Using physical models as a mapping layer between motion and sound provides intuitive and engaging interaction patterns, as physically plausible behaviors echo everyday experience. The dynamic nature of the mapping can strengthen engagement, but authoring such systems remains difficult because of their possible instability.

2.1.3 Towards Mapping by Example: Pointwise Maps and geometrical properties

Several authors proposed to specify continuous mapping functions from a discrete set of examples — i.e. pairs of motion-sound parameters. The continuous mapping surface is retrieved by interpolating between the presets with respect to particular geometrical properties (Bowler et al., 1990).

Goudeseune (2002) introduced a method for building static mappings with pointwise maps and interpolators. The mapping is defined by a set of unit relationships between input and output, that generalizes to a continuous mapping using piecewise linear interpolation. This approach reduces the construction of a continuous mapping to specifying finite set of points, and it guarantees the consistency of the mapping through continuity. Van Nort et al. (2004) extended this method to include stronger geometrical constraints. They use an interpolation scheme based on Regularized Spline with Tension to respect some desirable properties of the mapping: continuity, derivability, continuous higher derivatives and computational complexity. Similarly, Bencina (2005) present a system for audio control based on 2D interface. Recently, Van Nort et al. (2014) formalized this set of approaches through a topological and a functional view of mapping design, that accounts for musical context as a determinant in mapping analysis. Presets can be added as point on the interface and natural neighbor interpolation is used to dynamically interpolate between presets. Bevilacqua et al. (2005) implemented another strategy in the 'MnM' mapping toolbox, based on Single Value decomposition. It provides a way to build multidimensional linear mappings from a set of examples of input and output data.

> Methods based on interpolated pointwise maps reduces the creation of continuous mappings to specifying a set of input-output pairs. However, each *preset* must be specified manually rather than by continuous movements.

2.1.4 Software for Mapping Design

Number of domain-specific languages have become popular in the computer music community, using both textual[2] and graphical programming

2 See for example Supercollider (http://supercollider.sourceforge.net/) and ChucK (http://chuck.cs.princeton.edu/).

paradigms[3]. Several research groups developed software tools to facilitate mapping design. *MnM* is a mapping toolbox for Max/MSP using matrix representations for control parameters, gesture parameters, and their mapping. It implements various methods, from basic matrix operations to machine learning techniques such as PCA (Bevilacqua et al., 2005). Van Nort and Wanderley (2006) present the LoM mapping toolbox for quick experimentation with geometric methods in Max/MSP. The toolbox implements three interpolation methods : piecewise linear, multilinear, and Regularized Spline with Tension. Libmapper, introduced in Malloch et al. (2008) is a network-based mapping software that enables collaborative development and performance, which provides users with an interface for explicit mapping, embeds number of implicit mapping methods, and bridges to other tools such as MnM. CrossMapper implements a similar approach, with the addition of a graphical interface inspired by the Reactable (Liam et al., 2012). Other software are designed to achieve the same goals with specific design choices directed towards multi-touch systems (Kellum and Crevoisier, 2009), laptop music (Fiebrink et al., 2007) or novice users (Gelineck and Böttcher, 2012).

2.2 Mapping with Machine Learning

Machine learning is increasingly popular in the NIME community. We can identify two perspectives for the design of sonic interactions focusing on movement modeling and on multimodal modeling, respectively. The former approach exploits gesture recognition as a way to design both discrete and continuous interaction paradigms, that we review in Sections 2.2.2 to 2.2.4. The latter method utilizes regression techniques to learn continuous mapping between motion and sound parameters, and is reviewed in Section 2.2.5.

2.2.1 An Aside on Computer vision

As we specifically focus on mapping design, we do not give an extensive overview of input devices for musical expression. However, we aim to highlight computer vision as one of the primary use of machine learning in interactive computer music. Computer vision has been used for capturing movements of performers, with applications spanning from interactive installations to tangible interfaces for music control.

Number of works exploited computer vision for gesture capture, analysis and recognition in the fields of musical creation, interactive systems and musical gesture research (Camurri, 2000; Camurri et al., 2004). For example, Eyesweb is a computer vision software platform designed for performing arts, integrating specific image processing and gesture recognition techniques (Camurri et al., 2000a).

3 See for example Cycling'74 Max (http://cycling74.com/), PureData (http://puredata.info/).

Other approaches take advantage of computer vision systems for the creation of tangible interfaces. Some of these interfaces, developed in an academic context, have become popular to a wide audience. It is the case of the Reactable[4], a multi-touch tangible interface dedicated to musical creation and performance (Jorda et al., 2005; Jordà, 2008). More recently, several commercial devices such as Microsoft Kinect or Leap Motion, that integrate elaborate skeleton extraction methods, have been used in a variety of artistic works. Several recent research papers present applications of the kinect to gestural control of music, for example for instrument augmentation (Trail et al., 2012) or as a music controller (Sentürk et al., 2012).

2.2.2 Discrete Gesture Recognition

Before their application to the design of the mapping as such, machine learning algorithms have been successfully applied to gesture analysis, in particular in the context of conducting gesture analysis and recognition. Number of research projects have focused on capture, analysis and recognition of conducting gestures (Marrin and Picard, 1998; Lee and Nakra, 2004; Kolesnik, 2004). Notably, many gesture recognition techniques have been applied to real-time identification of the movements of the conductor, for example Hidden Markov Models (HMMs) (Kolesnik and Wanderley, 2005), Dynamic Time Warping (DTW) (Bettens and Todoroff, 2009), and Artificial Neural Networks (ANNs) (Bruegge et al., 2007).

From then, gesture recognition became a generic tool for the design of user-specific mappings. It makes possible the identification of particular gestures that might carry a semantic meaning, helping the implementation of metaphors for the control of audio processing. Since the first experiments in the 1990s (Sawada and Hashimoto, 1997), many techniques have been applied to gestural control of music.

Using a sensor glove, Modler (2000) proposed a system for gesture recognition based on time delay neural networks, which was later applied to recognition from video input (Modler et al., 2003). An implementation of Hidden Markov Models (HMMs) in Max/MSP is presented in Bettens and Todoroff (2009), with applications to gesture recognition. Recently, Gillian implemented various gesture recognition algorithms for musician-computer interaction in Eyesweb (Gillian, 2011; Gillian and Knapp, 2011). The toolbox integrates Naive Bayes for static pose identification, Dynamic Time Warping (DTW), Hidden Markov Models and Support Vector Machines (SVMs) — the latter model is also implemented in Wekinator Fiebrink (2011). Gillian and Paradiso (2014) later integrated these developments in the Gesture Recognition Toolkit, a c++ library and GUI for real-time gesture recognition. Recently, Gillian and Paradiso (2012) presented a system called *Digito* which uses a mixed discrete-continuous paradigm. A recognition algorithm is used to identify finger tapping from video input in order to trigger sounds. After triggering, an explicit mapping strategy is used to continuously modulate a FM synthesizer with hand movements.

4 Reactable: http://www.reactable.com/

> Many methods for real-time gesture recognition have been proposed in the NIME community. However, many uses of gesture recognition are confined to discrete interaction paradigms such as triggering a musical event when a particular gesture is recognized.

2.2.3 Continuous Gesture Recognition and Temporal Mapping

Several authors pointed out the importance of continuous interaction in DMIs and movement sonification. Recent approaches focus on extending gesture recognition and analysis methods to include continuous representations of gestures in terms of temporal and/or dynamic variations.

Bevilacqua et al. (2005, 2010) developed Gesture Follower for continuous gesture recognition and following. The system is built upon a template-based implementation of Hidden Markov Models (HMMs). It can learn a gesture from a single example, by associating each frame of the template gesture to a state of a hidden Markov chain. At runtime, the model continuously estimates parameters characterizing the temporal execution of the gesture. In particular, the system performs a real-time alignment of a live gesture over a reference recording, continuously estimating the *time progression* within the reference template.

The Temporal Mapping paradigm introduced by Bevilacqua et al. (2011) takes advantage of this system for continuously synchronizing audio to gestures. The results of the temporal alignment computed using *gesture follower* is mapped to a granular or phase-vocoding audio engine which realigns the audio track over the live performance of the gesture (see figure 2.1). Thus, modeling the accurate time structure of the gesture provides a new way of interacting with audio processing, mainly focused on the temporal dimension of the sound. However, the restriction to single-example learning limits the possibility of capturing the expressive variations that intrinsically occur between several performances of the same gesture.

Caramiaux et al. (2014a) extended this approach with an adaptive system based on particle filtering. Gesture Variation Follower (GVF) is a template-based method allowing to track 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. show that the model is efficient for early and continuous recognition. It consistently tracks gesture variations, which allows users to control continuous actions through gesture variations. The variations estimated by the system must be programmed as a specific state-space model, and therefore need to be adapted to each use-case.

> Recent approaches aim to overcome the limitations of the classification-triggering paradigm of gesture recognition, by implementing temporal models that characterize gestures as continuous processes varying in timing and dynamics. Most approaches, however, are restricted to learning from a single example.

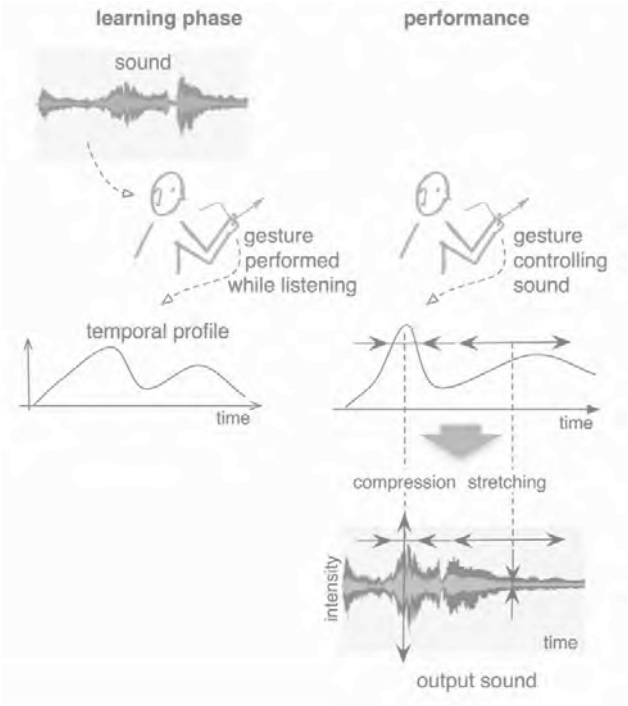


Figure 2.1: An overview of the temporal mapping strategy. During learning, a gesture is performed while listening to train the model. During performance, a new gesture is aligned on the reference to drive the time-stretching of the initial audio sample. From Bevilacqua et al. (2011), © Springer-Verlag Berlin Heidelberg 2011.

2.2.4 Multilevel temporal models

In *Gesture Follower*, each gesture is represented as a single time profile, and the gestures are independent during the recognition process. This assumption can be limiting: representing gestures as unbreakable units does not enable the creation of complex time structures in musical performance. Widmer et al., investigating artificial intelligence methods to analyze musical expressivity, underline the need for multilevel models: "Music performance is a multilevel phenomenon, with musical structures and performance patterns at various levels embedded in each other." (Widmer et al., 2003).

Recent findings about pianists' finger tapping emphasize two factors constraining musical gestures: biomechanical coupling and *chunking* Loehr and Palmer (2007). Introduced by Miller in the fifties Miller (1956), *chunking* suggest that "perceived action and sound are broken down into a series of chunks in people's mind when they perceive or imagine music" Godøy et al. (2010). More than just segmenting a stream into small entities, chunking refers to their transformation and construction into larger and more significant units. Jordà (2005, 2008) argued for the need of considering different control levels allowing for either intuitive or compositional decisions. Recently, Caramiaux (2012) highlighted the intricate relationship existing between hierarchical temporal structures in both gesture and sound when the gesture is performed in a listening situation. Thus, the design of sys-

tems implementing action-sound mapping should take into account different levels of temporal resolution, organized in a hierarchical structure.

Other fields of study such as speech processing (Ostendorf et al., 1996; Russell, 1993) and activity recognition (Aggarwal and Cai, 1997; Park and Aggarwal, 2004; Guerra-Filho and Aloimonos, 2007; Duong et al., 2009) exhibit a growing interest for hierarchical representations. Several extensions of Hidden Markov Models have been proposed to address its independence limitations. In the Segmental Hidden Markov Model (SHMM) (Ostendorf et al., 1996), each hidden state emits a sequence of observations, or *segment*, given a geometric shape and a duration distribution. The model has been successfully applied to time profile recognition of pitch and loudness (Bloit et al., 2010) and was exploited for gesture modeling in a recent study (Caramiaux et al., 2012). However, the model is not straightforward to implement for online inference. As an alternative, we proposed in previous work to use a hierarchical extension of HMMs with a structure and learning method similar to Gesture Follower. We developed and evaluated a real-time implementation of the Hierarchical Hidden Markov Model for the context of music performance (Françoise, 2011).

> Several movement modeling techniques integrate segmental and hierarchical representations, that provide a more consistent framework for gesture analysis. Nonetheless, they have not yet been fully exploited for designing the mapping between motion and sound.

2.2.5 Mapping through Regression

We presented mapping design techniques that draw upon gesture recognition and continuous movement models. Another thread of research considers supervised learning to directly learn the mapping between motion and sound parameters through regression.

Most approaches rely on neural networks, that have a long history in machine learning for their ability to learn the characteristics of non-linear systems. Lee et al. (1992) presented a system for simultaneous classification and parameter estimation from gestures to control parameters. They used feed-forward neural networks to control a virtual instrumentalist or a bank of sound generators. Similar systems were designed to learn the mapping between a data glove and a speech synthesizer (Fels and Hinton, 1993; Modler, 2000). Several generic implementations of neural networks have been proposed in PureData[5] (Cont et al., 2004) and in the *Wekinator*[6] (Fiebrink, 2011). Particular models can provide novel opportunities for mapping design. For example, Echo State Networks can be used to generate mapping stochastically, and their extreme non-linearity questions the boundary between control and *uncontrol* (Kiefer, 2014).

> Neural Networks can be very efficient for modeling non-linear systems, and are a powerful tool for mapping design. However, training such models can be tedious, notably because of the lack of transparency of the training process.

5 PureData: http://puredata.info/
6 Wekinator: http://wekinator.cs.princeton.edu/

2.2.6 Software for Mapping Design with Machine Learning

Many methods for discrete gesture recognition have been implemented as Max or Pure Data externals, and number of machine libraries are available online for different languages and platforms[7]. Among others, the SARC Eyesweb catalog[8] (Gillian and Knapp, 2011) and Gesture Recognition Toolkit[9] (Gillian and Paradiso, 2014) implement number of gesture classification algorithms (SVM, DTW, Naive Bayes, among others). The Wekinator[10] (Fiebrink, 2011) — detailed thereafter in Section 2.3.3 — implements a wide range of methods from the Weka machine learning toolbox, such as Adaboost, Neural Networks, and Hidden Markov Models. Several models for continuous gesture recognition and following are also available, such as Gesture Follower[11] (Bevilacqua et al., 2010) and Gesture Variation Follower (GVF)[12] (Caramiaux et al., 2014a).

2.3 Interactive Machine Learning

Today, every computer user constantly interacts with machine learning through search engines, spam filtering, recommender systems, or voice and gesture recognition. User interaction with machine learning algorithms is increasingly important. Integrating users in the learning process is a growing focus of several approaches at the intersection of the machine learning and Human-Computer Interaction (HCI) communities. Human intervention is becoming central in machine learning, either to provide rewards to an algorithm's actions (Knox, 2012), or to label a subset of data suggested by an active learning algorithm (Settles, 2009). Recent studies focus on pedagogical approaches that emphasize how explaining machine learning to users helps improving user adaptation methods (Kulesza, 2014).

In this section, we focus on the particular thread of this interdisciplinary research called Interactive Machine Learning (IML). IML aims to integrate users at all steps of the learning problem, from the creating of the training examples to the training and evaluation of the models.

2.3.1 Interactive Machine Learning

Interactive Machine Learning (IML) is a subdomain of HCI that investigates how to make machine learning more usable by end users, both to support human interaction and to improve learning tasks through human intervention. The term was first proposed by Fails and Olsen (2003) who introduced a novel workflow for user interaction with machine learning algorithms. Fails and Olsen argue that in classical machine learning, the

7 The *Machine Learning Open Source Software* website currently indexes more than 550 entries: http://mloss.org/software/
8 http://www.nickgillian.com/software/sec
9 https://github.com/nickgillian/grt
10 Wekinator: http://wekinator.cs.princeton.edu/
11 Gesture Follower: http://ismm.ircam.fr/gesture-follower/
12 GVF: https://github.com/bcaramiaux/ofxGVF

training data is always fixed and user can only intervene by evaluating the results of the training. Their proposal integrate users at all steps of the process, from providing training data to training and evaluate machine learning models. They propose to move the focus on creating data that will correct the classifier. They illustrate their approach with the Crayons application, that involve users in iteratively creating data, training, and evaluating an image classifier.

Fogarty et al. (2008) presented Cueflik, a system for web-based image search where images are ranked according to a set of user-defined rules on image characteristics. The rules are learned through a classification as positive or negative examples of the target concept. Other approaches investigate visualization to support the design of classifiers (Talbot et al., 2009), or propose to improve programming environment through a better integration of the algorithm and data (Patel et al., 2010).

> Interactive Machine Learning investigates how user intervention can improve the design of machine learning systems. Several work highlight the users' efficiency in building classifiers through the iterative specification and evaluation of training examples.

2.3.2 Programming by Demonstration

Programming-by-Demonstration is a related field of computer science that studies tools for end-user programming based on demonstrations of the target actions. One of the primary challenges in Programming-by-Demonstration (PbD) is to go beyond this simple reproduction of actions to the generalization of tasks or concepts.

Hartmann et al. (2007) highlight the difficulty for interaction designers to map between sensor signal and application logic because most tools for programming interaction are textual and are rarely conceived to encourage rapid exploration of design alternatives. They introduce the Examplar system that provides a graphical interface for interactive visualization and filtering of sensor signal, combined with pattern recognition based on Dynamic Time Warping (DTW). A qualitative user study shows that their approach reduces the prototype creation time and encourages experimentation, modifications and alternative designs through direct user experience assessment. A similar approach is adopted by Lü and Li (2012) for the case of multitouch surface gestures.

Programming-by-Demonstration has become an primary methodology in robotics, as way to teach robot interactions from human examples. This body of work, in particular its thread focusing on robot motor learning, is described later in Section 2.5.3.

The PbD methodology has been applied to the field of interactive computer music. Merrill and Paradiso (2005) described a PbD procedure for programming musical interaction with the FlexiGesture interface. The system implements a classification-triggering paradigm for playing sound samples, along with continuous mappings where only the range of the input sensor is adapted to the range of the parameter.

Programming-by-Demonstration allows users to specify tasks and concepts to computer systems through a set of examples. It has been applied to interaction design, notably in music, highlighting its benefit for evaluating multiple design in a quick interaction loop. However, most works focus on either discrete paradigm based on gesture classification, or direct mapping of filtered sensor streams to software actions.

2.3.3 Interactive Machine Learning in Computer Music

Our review of mapping strategies in Digital Musical Instrument design highlights that in sound and music computing, complex mappings involving many-to-many associations are often more preferred to simple triggering paradigm. However, most approaches to interactive machine learning have focused on discrete tasks such as classification.

To overcome such issues and fit the context of interactive computer music, Fiebrink's Wekinator (2011) implements various machine learning methods for both recognition and regression in a user-centered workflow illustrated in Figure 2.2. The Wekinator 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.

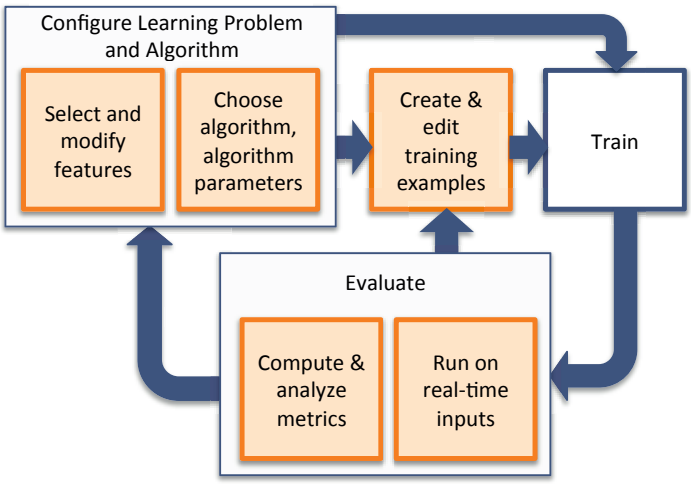


Figure 2.2: The Interactive Machine Learning workflow of the Wekinator. Source: Fiebrink (2011), © Copyright by Rebecca Anne Fiebrink, 2011.

Through three studies with students in composition and professional composers, Fiebrink et al. (2011) showed that users consistently iterate over designs, analyzing errors and refining the training data and algorithms at each step. This work emphasizes that taking into account users' needs in machine learning is crucial to get efficient and expressive designs. For example, while evaluation methods such as cross-validation are widespread in the machine learning community, users often prefer direct evaluation — i.e. interaction with a trained model. The third study is a case study that aimed at building a gesture recognizer for a professional cellist. The

study shows that the actual recognition rate does not matter as much, from the user's perspective, as the shape and smoothness of the classification boundary defining the transition between several 'modes'.

The approach of Wekinator was further extended by Laguna and Fiebrink (2014) with 'Gesture Mapper', that implements a generator of mapping alternatives and a history tree encoding the iterative design process.

> Interactive Machine Learning is an efficient and expressive tool for interactive computer music. Integrating users in the learning process improves the design of classifiers and regression models for music performance and allows novice users to quickly and efficiently iterate through design alternatives.

2.4 Closing the Action-Perception Loop

In this section, we outline contemporary theories in embodied music cognition and sound perception, and report on recent studies investigating the involvement of the body in sound perception. We motivate our approach of MbD from the mapping through listening design principle and interactive machine learning techniques.

2.4.1 Listening in Action

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. Anderson (2003), reviewing the field of embodied cognition in 2003, reports four aspects of embodiment: physiology, evolutionary history, practical activity, and sociocultural situatedness.

Commenting on Merleau-Ponty, Anderson (2003) notes that

> perception and representation always occur in the context of, and are therefore structured by, the embodied agent in the course of its ongoing purposeful engagement with the world. Representations are therefore 'sublimations' of bodily experience, possessed of content already, and not given content or form by an autonomous mind.

O'Regan and Noë (2001) further push the idea of perception as an active phenomenon, considering that sense organs are themselves dynamic instruments of exploration. Perception is therefore intrinsically an active phenomenon, that can only be realized through action in an environment.

Several authors argue that embodiment is also essential at higher cognitive levels. According to Lakoff and Johnson (1999), "The same neural and cognitive mechanisms that allow us to perceive and move around also create our conceptual systems and modes of reason. Thus, to understand reason we must understand the details of our visual system, our motor system, and the general mechanisms of neural binding".

The theory of embodied cognition has a significant impacts on current trends in Human-Computer Interaction research (Kirsh, 2013), and interaction design (Dourish, 2004).

In *Embodied Music Cognition and mediation technology*, Leman (2008) underlines the importance of "corporeal engagement" in music experience. Leman argues that music originates from body movements that convey the inner intent of the performer, such as emotion. Leman suggest that listeners engage with musical listening through motor simulation, putting bodily experience as a primary vector of musical expression.

MOTOR THEORY OF SOUND PERCEPTION

Recent results in neuroscience support a "motor theory of sound perception" which suggest that motor images are integral to sound perception (Kohler et al., 2002).

According to Godøy (2003), "we mentally imitate the sound-producing action when we attentively listen to music, or that we may image actively tracing or drawing the contours of the music as it unfolds". This concept is supported by recent experiments investigating spontaneous motors responses to sound stimuli (Godøy et al., 2006a,b; Caramiaux et al., 2010a, 2014b).

Several author highlight the importance of preserving or simulating an energy transfer from movement to sound (Leman, 2008; Caramiaux et al., 2014b). Depending on the context, however, we observe a wide range of strategies for associating gestures to sound, such as mimicking the sound-producing actions (Godøy et al., 2006b), or tracing the perceived properties of the sound (Godøy et al., 2006a). Recently, Caramiaux et al. (2014b) showed that the identification of the sound source is decisive in gestural strategies. While identified sounds often induce gestures that mimic the sound-producing actions, the lack of identification of the physical source leads to tracing the sound properties.

Most importantly, all authors report a large variety of strategies. Associations between gestures and sound are highly idiosyncratic, suggesting that systems for designing such interactions should be able to adapt to the variability induced by contextual, cultural, and personal factors.

2.4.2 Generalizing Motion-Sound Mapping

Mapping has often been defined as the layer connecting motion sensors and sound synthesis parameters (Rovan et al., 1997). This definition is obviously advantageous from a technical point of view, as it is restricted to a set mathematical operation between parameters. However, sound control parameters might not always be musically, perceptually or metaphorically relevant, nor sensors or their extracted motion features might be relevant to movement perceptive or expressive attributes. Elaborate sound synthesis models, such as physical models (Rovan et al., 1997), already integrate a significant amount of translation of the input parameters — e.g. the speed and pressure of a bow, — to musically relevant parameters such as pitch, loudness, etc.

Obviously, describing the mapping alone is not sufficient to understand the implications in terms of the action-perception loop, and it should systematically be described in conjunction with the movement input device, sound synthesis, and all other intermediate operations (pre-processing, feature extraction, parameter mapping). Recently, Van Nort et al. (2014) proposed topological and functional perspectives on mapping that argue for considering the musical context as a determinant in mapping design:

> [...] we must remember that mapping per se is ultimately tied to perception and, more directly, to intentionality. In the former case this means building a mapping around certain key action–sound associations through the design of appropriate correspondences of system states. In the latter case, this means conditioning this association towards the continuous gestures that will ultimately be experienced. (Van Nort et al. (2014))

This brings out the alternative perspective of considering mapping as the entire action–perception loop that relates the performer's movements to the resulting sound.

MAPPING THROUGH LISTENING

We recently formalized a design principle we call "Mapping through Listening", that considers listening as the foundation and the first step of the design of the relationships between motion and sound (Caramiaux et al., 2014c).

Our approach 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 a design principle that considers embodied associations between gestures and sounds as the essential component of mapping design.

2.4.3 Motivating Mapping-by-Demonstration

Our overview of the related work highlights a shift 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. Nevertheless, to our knowledge, no general framework has yet been proposed to explicitly integrate both approaches.

In this thesis, we formulate a general framework, termed Mapping-by-Demonstration (MbD), that intersects the approaches of mapping through

listening and interactive machine learning. We consider jointly performed movements and sounds that express embodied associations as primary material for interaction.

Our framework considers listening as a starting point for the design of the mapping. We propose to learn the mapping from a set of *demonstrations*, that make explicit the relationship between motion and sound as an acted interaction. We use joint recordings of gestures and sounds to learn a mapping model using statistical modeling. Users are therefore embedded in an interactive machine learning design loop that alternates the creation of training examples and the evaluation of the designed motion-sound relationship.

2.5 Statistical Synthesis and Mapping in Multimedia

We presented how the technological and artistic perspective on motion-sound mapping evolved from direct wiring between motion sensor parameters and sound synthesis parameters, to approaches taking advantage of an intermediate model of interaction. As Interactive Machine Learning evolves in the music community, we think important to draw knowledge in cross-modal modeling from other fields of study. In this section, we review the recent approaches to cross-modal mapping in speech and motion processing to motivate the use of a probabilistic modeling approach.

Note: This section discusses technical aspects of HMM-based speech recognition, synthesis and mapping, for which we assume basic knowledge of HMMs. The formalism of HMMs is further reviewed in Chapter 4.

2.5.1 The case of Speech: from Recognition to Synthesis

AUTOMATIC SPEECH RECOGNITION

In this section, we make a detour through the speech processing community to highlight how the use of generative sequence models expanded from recognition problems to synthesis and mapping. Although Automatic Speech Recognition (ASR) systems often contain a *front-end* component relating to language modeling, we focus on the *back-end* part which is interested with lower-level acoustic modeling (Taylor, 2009); for extensive overview of the machine learning paradigms in speech recognition, see Deng and Li (2013). While Neural Networks were the popular model until the 1990s, Hidden Markov Models (HMMs) and their extensions have dominated ASR for almost 20 years. HMMs combine a latent model with Markov dynamics, that consistently encodes temporal variations, with a continuous observation model, generally a Gaussian Mixture Model (GMM) that measures the fit to the observed acoustic features (Gales and Young, 2007). While recent developments in Deep Neural Networks allowed to critically improve the observation models of HMM recognizers, Gaussian Mixture Models (GMMs) still represent an excellent choice for acoustic modeling, as noted by Hinton et al. (2012):

> GMMs have a number of advantages that make them suitable for modeling the probability distributions over vectors of input features that are associated with each state of an HMM. With enough components, they can model probability distributions to any required level of accuracy, and they are fairly easy to fit to data using the Expectation-Maximization (EM) algorithm.

Most importantly, the computational cost of training and evaluating GMMs is fairly low, whereas training deep neural networks remains computationally very expensive. Moreover, in cases the training set is very small, elaborate training methods for GMM are competitive with deep learning techniques (Hinton et al., 2012). Finally, the acoustic model of HMM-GMM models can be flexibly adapted, for example to change the speaker characteristics. This flexibility has been driving research in HMM-based sound synthesis for two decades.

STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Although to date, corpus-based concatenative synthesis produces the highest quality Text-To-Speech (TTS), it is both memory expensive and by nature hardly adapts to speaker, affect, or expressive variations — it would require massive databases and would be extremely time consuming. The need for controlling speech variations led to the development of so called statistical parametric speech synthesis, that models acoustic parameters using a stochastic generative model. We give here a few insights into the recent developments of HMM-based speech synthesis, for a recent review, see for example Tokuda et al. (2013).

The rapid development of statistical speech synthesis is due to the well-established machine learning method from ASR. As a matter of fact, HMMs provide a unified modeling framework for speech analysis and synthesis, allowing to transfer methods — such as speaker adaptation, — from recognition to synthesis. However, Dines et al. (2010) highlight a gap between ASR and TTS: while speech modeling is unified through generative models, the two fields present important differences in implementation. As the goal is to reconstruct instead of discriminate, different features are used for synthesis (usually F_0, MFCC, and their derivatives); the models often use different topologies (number of states and Gaussian components); and, critically, synthesis methods require explicit duration modeling for consistently reconstructing sequences of phonemes (Dines et al., 2010).

The typical setup of a TTS system is presented in Figure 2.3. The acoustic waveform is synthesized by combining an excitation signal with spectral filtering, by analogy with the human speech production process. Except for this particular representation of the speech signal, the training part of the system is closely related to those of ASR. The driving question, from our perspective, relates to parameter generation: *how to generate smooth and 'natural' acoustic feature sequences from a discrete-state model?*

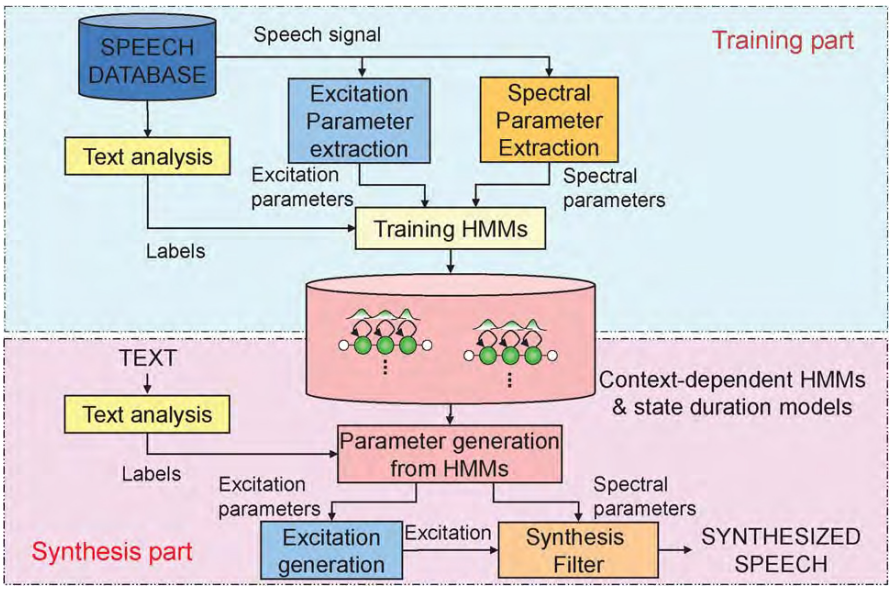


Figure 2.3: Overview of the HMM-based Speech Synthesis System. From Tokuda et al. (2013), © 2013 IEEE.

PARAMETER GENERATION ALGORITHMS

Tokuda et al. (2000) propose several methods for generating the optimal parameter sequence given a model. Maximum Likelihood Parameter Generation (MLPG) consists in estimating the optimal sequence of hidden states using explicit duration modeling with Hidden Semi-Markov Model (HSMM). In a second step, the parameters are estimated by maximizing the conditional probability of the observation sequence given the hidden state sequence. To ensure the consistency and continuity of the synthesized trajectory, dynamic features (delta and delta-delta features) are explicitly related to static features for generation. The main drawback of this approach is the strong reliance on a fixed state sequence, which may dramatically propagate errors from the state sequence to the generative features. Alternatively, Tokuda proposes a method based on the EM algorithm that directly maximizes the likelihood of the observation sequence given the model. Zen et al. (2007) address the inconsistency between static and dynamic feature during training, reformulating the HMM with dynamic feature constraints as a *trajectory* HMM. (Wu and Wang, 2006) took advantage of this formalism to introduced a Minimum Generation Error training algorithm that minimizes the synthesis error.

ADVANTAGES AND LIMITATIONS OF HMM-BASED SPEECH SYNTHESIS

According to Tokuda et al. (2013), the main advantages of HMM-GMM synthesis are its robustness, small footprint and flexibility in changing speaker characteristics. The power of statistical speech synthesis resides in the flexibility of the approach, that can change speaker characteristics, integrate expressive features and facilitate multilingual synthesis (Tokuda et al., 2013). Notably, models can be adapted quickly to a new speaker in an unsupervised — and possibly incremental

— way. Maximum Likelihood Linear Regression (MLLR), the most popular method for speaker adaptation, is based on a linear transform of Gaussian components' mean and covariance (Leggetter and Woodland, 1995; Yamagishi et al., 2009). The introduction of articulatory, prosodic or affective features can improve expressive qualities of the synthesized speech.

The major drawback, however, is the quality of the synthesized speech. This is mostly due to the quality of the source-filter and vocoder models, but *oversmoothing* (or *undershooting*) also significantly decrease the speech naturalness. Toda and Tokuda (2007) addressed the latter problem using a global variance method that penalizes feature sequences with low variance. As for ASR, recent advances in deep neural networks make them promising for speech synthesis. Zen et al. (2013) highlight the advantages of deep neural networks that can integrate feature extraction, have distributed representation, and implement complex hierarchical structures. The authors note, however, that their current implementation is much less efficient than classical HMM-based approaches.

Most of the methods proposed for speech synthesis are oriented towards applications in TTS that use offline inference and generation. Recently, Astrinaki et al. (2012a) proposed a reactive implementation of the HTS speech synthesis system. The system can generate speech with low latency through reducedd phonetic context and generation of short-term speech parameter trajectories. The parameter trajectories are generated using MLPG on a sliding window. This method was implemented in the MAGE software that also extends this approach to continuous control of higher-level speech or singing-voice parameters (Astrinaki et al., 2012b).

> Statistical parametric speech synthesis provides a flexible framework for expressive synthesis, that can integrate prosodic, articulatory of affective features. Several methods have been proposed for efficient and robust parameter generation, but most of them are oriented towards offline generation. Recent approaches implement reactive speech synthesis that allows for low-latency parameter generation.

Novel applications such as speaker conversion or automated translation are encouraging a shift from pure synthesis towards the issue of sequence mapping. Voice conversion is typically performed using GMMs, which are suitable as the input and output sequence belong to the same modality (Toda et al., 2007). For the complex relationships that exist in cross-modal mapping, more elaborate models based on sequence modeling are desirable, as discussed in the next section.

2.5.2 Cross-Modal Mapping from/to Speech

We now review the body of work that deals with cross-modal mapping where speech is used to drive movement generation. We first consider the case of acoustic-articulatory inversion, and then the more general problem of speech-driven character animation.

ACOUSTIC–ARTICULATORY INVERSION

Acoustic-articulatory mapping — also known as *speech inversion* — aims at recovering from acoustic speech signals the articulation movements related to vocal production.

As for speaker conversion, the initial approach draws upon the use of GMMs for mapping. Toda et al. (2004) proposed to use the GMMs for regression, following the theoretical work of Ghahramani and Jordan (1994). 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. Toda et al. (2004) evaluated synthesis methods, respectively using static features or their combination with dynamic features. The model using dynamic feature constraints perform better in terms of Root Mean Square (RMS) error, however it requires to solve for the entire sequence, which is incompatible with real-time conversion. Richmond further extended the method by combining Gaussian mixtures with an artificial neural network (Richmond, 2006).

Subsequent work applied HMM to the problem of feature mapping: Zhang and Renals (2008) extended the trajectory HMM to learn a joint model of acoustic and articulatory features. Using the state sequence estimated from acoustic information, articulation movements are generated with dynamic constraints. Zhang and Renals showed that training with the trajectory HMM significantly reduces the RMS error compared with Maximum-Likelihood Training. Hueber and Badin (2011) and Zen et al. (2011) used a similar approach, that applies the trajectory model to HMMs and GMMs.

SILENT SPEECH INTERFACES

The purpose of silent speech interfaces is to synthesize speech from articulatory movements, with accessibility as primer application. They represent the direct mapping problem with respect to acoustic-articulatory inversion, and use the same set of methods. Analogously to the previous work, Toda et al. (2008) applied Gaussian Mixture Regression using both the Least Squares Estimate (LSE) criterion and Maximum Likelihood (ML) Estimate with dynamic feature constraints, showing that the trajectory model outperforms the standard LSE method. Hueber and Badin (2011) further extended this method to trajectory HMMs for generating acoustic speech from combined ultrasound imaging and video analysis. A perceptual study showed that HMMs outperform GMMs, but is still limited by a low recognition rate.

SPEECH–DRIVEN CHARACTER ANIMATION

Embodied conversational agents aim to create rich multimodal interactions between a human and a virtual character. Although language processing and speech communication are obviously central to the design of a realistic behavior, recent research focuses on non-verbal communication through body movement. Recent approaches to animation intend to make a link between verbal and non-verbal behavior by mapping acoustic speech to lip, face and body movements. As for the previous fields of study, initial approaches have focused on learning a static

relationship from acoustic features to motion features — e.g. from recognized phonemes to lip poses, — using codebook approach, neural networks or Gaussian Mixture Models (GMMs) (Rao et al., 1998; Chen, 2001). However, considering the dynamic properties of both speech and motion, many recent approaches are based on multimodal variations of HMMs, in particular to tackle the problem of non-uniqueness of the mapping between speech and motion (Ananthakrishnan et al., 2009).

Brand (1999) introduced a remapping strategy for HMMs to address the problem of speech driven face animation. They propose to train a HMM on the gesture features, that is then remapped — i.e. re-trained — and synchronized to the audio track. The face sequence is then generated based on the optimal hidden state sequence estimated on new audio. Other approaches focus on learning a multimodal HMM by concatenating speech and motion features in the observation vector during training. Then, synthesis is performed by evaluating the observation sequence of motion features corresponding to the state sequence estimated on speech features only.

Busso et al. (2005) exploited this method for the generation of head gestures based on acoustic prosodic features. An important disadvantage of the approach is the direct dependency of the synthesis on the Viterbi algorithm used to compute the sequence of Hidden states. In presence of noise, errors during the recognition process can lead to inconsistencies between speech and visual synthesis. To overcome this limitation, Yamamoto et al. (1998) proposed an EM-based approach, which is extended in Ding et al. (2013); Ding (2014) by a parametrization of the observation models of the motion features on emotion parameters.

HMM inversion, introduced in Choi et al. (2001), also addresses this issue by exploiting a Maximum Likelihood estimation of the visual parameters. Among the three cross-modal HMMs compared in Fu et al. (2005), HMM inversion shows the best results in comparison with remapping techniques. Alternatively, Li and Shum (2006) proposed the use of input-output HMMs in which the observations are conditioned on input variables (Bengio, 1996). While their results indicate a more accurate estimation of the visual parameters, the training procedure is very expensive, as the probability distributions over input variables must be learned using neural networks.

> Several methods for mapping between speech and movement take advantage of HMM-based synthesis techniques. Often, the parameter sequences are generated under dynamic constraints using a fixed state sequence estimated with the Viterbi algorithm on the input modality. Although alternative methods use the EM algorithm to maximize directly the likelihood of the output sequence given then model, most approaches remain incompatible with real-time continuous mapping.

2.5.3 Movement Generation and Robotics

ANIMATION Character animation followed two paths of rule-based and data driven approaches to movement synthesis. We do not aim to give a comprehensive review of statistical approaches to movement synthesis. However, we want to emphasize that probabilistic sequence models such as HMMs have been initially used for gesture recognition, and were later exploited for movement generation. Tilmanne (2013) gives a comprehensive overview of such methods.

As for speech synthesis, statistical Gaussian models have proved efficient for encoding the dynamics of human motion, and flexible for generating movement according to external variables such as style. Brand and Hertzmann (2000) proposed 'style machines', a method for extracting style parameters implicitly at training, that can then be used during generation for interpolating between styles.

Recently, Tilmanne et al. (2012) proposed a method for style adaptation in walking motion synthesis, inspired by speaker adaptation methods in speech processing. The method uses an explicit time model for synthesis, and is based on the training of an average model of walking, namely a Hidden Semi-Markov Model (HSMM). The average model can then be adapted on few examples of walking with a particular style. This approach was extended to style interpolation in Tilmanne (2013), where intermediate styles can be generated by creating a weighted mixture of models with different styles.

> Hidden Markov Models have been successfully applied to realistic motion synthesis. The flexibility of the models gives the possibility to encode various parameters of motion, such as style. However, with respect to our field, the process is often not interactive, as training and sequence generation are often performed offline.

ROBOTICS In robotics, computational movement modeling techniques evolved from early rule-based methods to data-driven approaches to movement generation. Inspired by theories of human motor learning, robotics is experiencing a sustained trend interested in learning approach by *imitation* of human behavior (Schaal, 1999). In this section, we describe several methods for movement learning based on a Programming-by-Demonstration (PbD) methodology, that integrate human teachers as a fundamental aspect of robots motor learning.

Argall and Billard (2011) define Learning from Demonstration (LfD) as learning policies — mappings between states of the world and action — from demonstrations, or examples, provided by a teacher. LfD mostly focuses on supervised learning from training data composed of state-action pairs, in opposition to reinforcement learning approaches that are more oriented towards learning from experience. Argall and Billard highlight the issue of *correspondence* between the teacher and the learner that is defined with respect with two mappings, as illustrated in Figure 2.4:

- the *Record Mapping* between the teacher's movement and the actual recording.

- the *Embodiment Mapping* between the dataset and the learner's observation/execution.

In the following, we consider both mapping as identity, where the demonstration is performed either by a human teacher equipped with sensors matching those of the learner, or by direct manipulation of the robot by a human operator. In Section 3.4, we propose a definition of the correspondence problem in the Mapping-by-Demonstration framework.

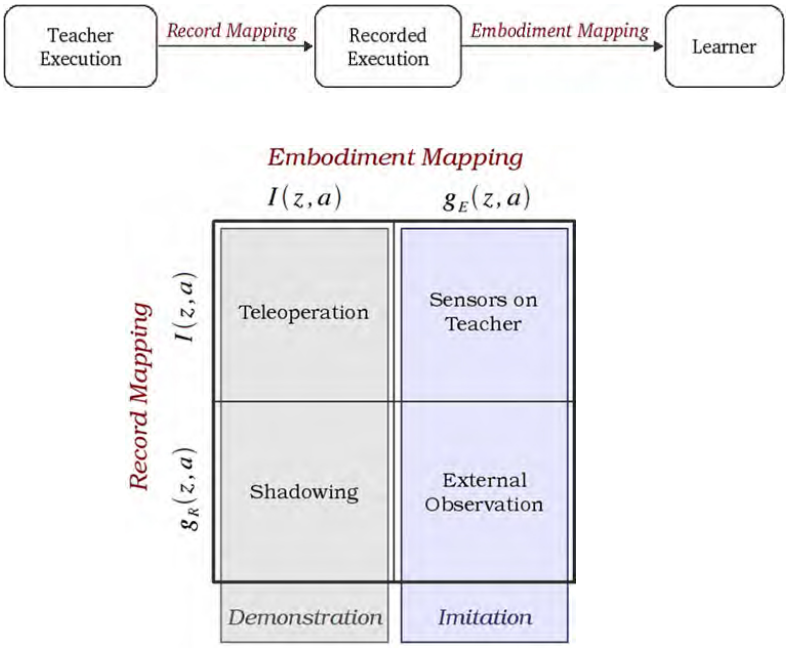


Figure 2.4: *Correspondence*: Mapping a teacher to a learner in a Learning from Demonstration paradigm. From Argall et al. (2009), © 2009 with permission from Elsevier.

We consider the specific case of motor learning by demonstration, where a robot learns a set of motor tasks from human demonstration. While many approaches are based on dynamical systems — see in particular Dynamic Movement Primitives (Schaal et al., 2000, 2003; Ijspeert et al., 2013), — several methods draw upon probabilistic models for motor task modeling, reproduction and generalization.

Calinon et al. (2007) proposed to learn a joint time/motion GMM from multiple demonstrations of a motor behavior. Demonstrations are realized by an expert teacher that directly manipulate the robot to show several variations of a task. Principal Component Analysis (PCA) is used to define a subspace of the motion parameters, and the authors proposed to use DTW to realigned the sequences. The reproduction and generalization of the tasks is realized through Gaussian Mixture Regression (GMR), that estimates the time profile of motion parameters from an input time vector. The authors showed the efficiency of the approach on robot motor imitation tasks, and later proposed a probabilistic representation of Dynamic Movement Primitives using GMR (Calinon et al., 2012).

Calinon et al. (2010) further extended the method by combining Hidden Markov Models (HMMs) with GMR where the weight of each Gaussian are estimated by a forward algorithm.[13] Moreover, the authors combined the predictions of the model with a *stabilizer* term ensuring dynamic constraints, which highlights the flexibility of the method for combining several constraints. The method was shown to outperform the time-based GMR, and gives similar results as Dynamic Movement primitives, but can more easily learn from several demonstrations with *variations*. Recently, the method was further extended to HSMMs (Calinon et al., 2011).

Note that while most method in speech synthesis are offline, motor control in robotics must be performed online, and both methods based on GMMs and HMMs can be computed in real-time. Most interestingly, recent developments based on GMR highlight how the transparency of the model makes it possible to extend the model's capabilities, for example by parameterizing the mean and covariance parameters of the Gaussians over specific tasks (Calinon et al., 2014).

> Robot Programming-by-Demonstration is a rich field of study that focuses on movement acquisition, learning and generalization. The approach explicitly integrates users in the machine learning process, where robot learning relies on the interaction with a human teacher. Several approaches use probabilistic sequence models for encoding and generalizing motor tasks, where motion synthesis is performed in real-time, and possibly conditioned on contextual factors.

2.5.4 Discussion

The fields of speech and motion processing provide us with a rich background on the analysis, mapping and synthesis of feature sequences. In both communities, the interest in statistical models expanded from classification and recognition to problems of synthesis. Most interestingly, several threads of research intersect these fields and focus on mapping between different modalities. Acoustic-articulatory inversion, speech-driven animation or silent speech interfaces aim to map between feature sequences representing speech and motion.

Many current methods use statistical models for performing such a mapping, either using Gaussian Mixture Models (GMMs) for regression (also called Gaussian Mixture Regression (GMR)), or sequence models such as Hidden Markov Models (HMMs). Both models are flexible and can be adapted to a number of factors, such as style, expressiveness, context, or new users.

Nonetheless, several essential differences exist between this body of research and our field. First, both in acoustic-articulatory inversion and speech-driven animation, the relationship between speech and motion is relatively well defined in that it emerges from the same physical process. This makes it possible to train statistical models on extensive large databases that might represent a wide range of variations between users, contexts or styles. On the contrary, we aim to address the very design of the relationship between motion and sound. Our goal is to let users define this relationship by di-

13 In the following, we call this method Hidden Markov Regression (HMR) (see Section 6.2)

rect demonstration. A challenge is therefore to implement recognition and mapping models in an interactive machine learning environment to be usable by a variety of users.

Second, we aim to continuously control sound from input movements: the generation of the sound parameters must be performed instantaneously, each time a new frame of motion parameters is available. Therefore, offline methods for synthesis and mapping cannot be applied in our case, and we need to develop new ways to perform inference in real-time.

2.6 Summary

This chapter discussed the related work in designing the relationship between motion and sound. We detailed the relevant approaches in mapping design that draw upon intermediate models of the mapping between movement and sound. Specifically, we reviewed how machine learning techniques and the emerging field of interactive machine learning can support creativity in music and sonic interaction design.

We related this technological perspective to theoretical approaches that consider the action-perception loop as a fundamental principle in the design of interactive systems. We motivated our framework of Mapping-by-Demonstration as the intersection between the mapping through listening methodology and interactive machine learning.

Finally, we reviewed the current trends in related fields of multimedia processing to motivate the use of probabilistic models for recognition, mapping and generation.

3

Mapping by Demonstration

This chapter formalizes the concept of Mapping-by-Demonstration (MbD). We start by motivating the approach with regards to the related work in mapping design based on embodied action-sound associations, and in interactive machine learning for computer music. Then, we define and describe Mapping-by-Demonstration, and we propose an architecture along with several desirable modeling properties.

3.1 Motivation

Our overview of the related work in the field of music computing highlights a transition from explicit definitions of motion-sound mapping to more implicit models. We review the two perspectives of mapping through listening and interactive machine learning, that respectively address a mapping design principle driven by listening and a framework for data-driven design of sonic interactions.

MAPPING THROUGH LISTENING

Our approach leverages on a body of work in the *{Sound Music Motion} Interaction* team at Ircam. In particular, Caramiaux (2012) studied the relationships between gesture and sound in musical performance from two perspectives. The first view investigates gestural responses to sound stimuli (Caramiaux et al., 2014b), and the second approach consists in developing motion modeling tools for interaction design (Caramiaux and Tanaka, 2013; Caramiaux et al., 2014a).

We recently formalized an approach to mapping design called *mapping through listening*, that considers listening as the starting point for designing the mapping between movement and sound in interactive systems (Caramiaux et al., 2014c). Mapping through listening is a design principle that considers embodied associations between gestures and sounds as the essential component of mapping design.

Mapping-by-Demonstration (MbD) is a framework for designing sonic interactions that draws upon this general principle. We propose to explic-

itly consider corporeal demonstrations of such embodied associations as a basis for learning the mapping between motion and sound.

INTERACTIVE MACHINE LEARNING

Recent developments in Interactive Machine Learning (IML) are bringing elaborate tools for designing by example to end users with varying expertise. The goal of Interactive Machine Learning (IML) is twofold: improving machine learning through user intervention, and empowering users with elaborate methods for interaction design.

Fiebrink (2011) particularly contributed to foster this approach in interactive computer music. Fiebrink's approach focuses on improving end-user interaction with machine learning, by integrating users' decisions at several steps of the process: editing of the training examples; model selection, tuning and training; evaluation through analytical results and direct interaction (Fiebrink et al., 2011).

A particularly interesting methodology is that of *play-along* mapping, introduced by Fiebrink et al. (2009). In play-along mapping, a *score* of sound presets is used as a guide to the definition of the training examples — e.g. performing gestures while listening. The approach, however, might require to define the score manually, and does not explicitly considers listening and perception as a starting point.

MOTIVATION: CLOSING THE ACTION-PERCEPTION LOOP

We propose to combine the design principle of mapping through listening with interactive machine learning in a framework we call Mapping-by-Demonstration (MbD). Our approach exploits interactive machine learning for crafting sonic interactions from embodied demonstrations of the desired motion-sound mapping.

3.2 Definition and Overview

3.2.1 Definition

We propose to define Mapping-by-Demonstration (MbD) as follows:

> **Definition:**
> Mapping-by-Demonstration (MbD) is a framework 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-by-Demonstration refers to the very active field of Programming-by-Demonstration (PbD) in robotics (Argall et al., 2009).[1]

1 Note that *imitation* learning is also widely used in robot motor learning (Schaal, 1999; Schaal et al., 2003). 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.

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 the role of embodied demonstrations for crafting movement control strategies. We draw upon the mapping through listening methodology we previously formalized, and further integrate the action-perception loop as a fundamental component of the design process through the use of Interactive Machine Learning.

3.2.2 Overview

We now give an overview of the workflow of the Mapping-by-Demonstration framework from a user's perspective, as illustrated in Figure 3.1. The framework implements an interaction loop iterating over two phases of *Demonstration* and *Performance*.

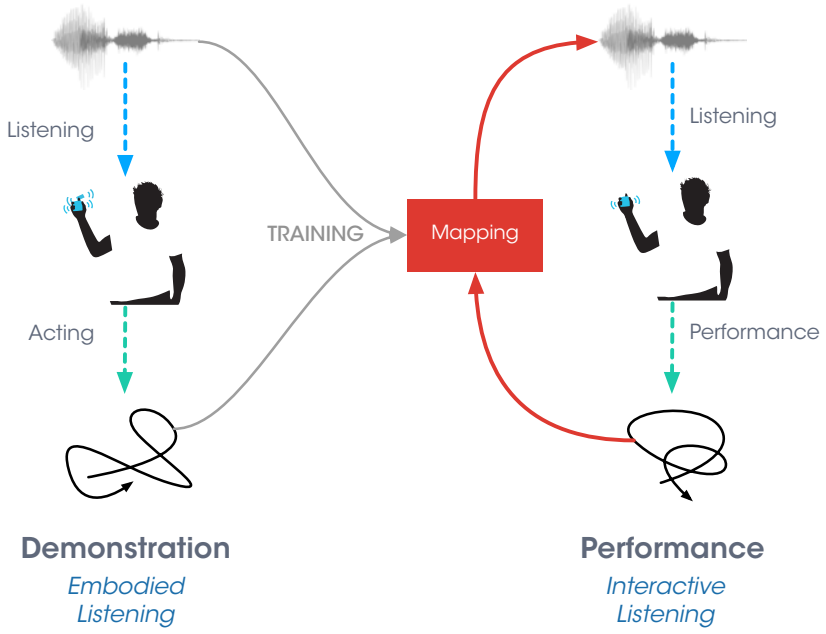


Figure 3.1: Overview of Mapping by Demonstration. Blue and green dashed arrows respectively represent listening and movement (as physical action). In *Demonstration*, the user's movement performed while listening is used to learn an interaction model, that continuously generates sound from new movements in *Performance*.

The demonstration phase starts with embodied listening where the user imagines a movement to associate with the sound (**listening**). Then, the imagined association between motion and sound needs to be acted to provide the system with an exemplar movement performed along the example sound (**acting**). We synchronously record the motion and sound parameter streams to form a joint motion-sound sequence that constitutes a demonstration. 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. Users can edit and annotate the training examples, select and adjust the parameters of the machine learning model. Once trained, this mapping model can be used in the *Performance* phase. The user can therefore 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. This framework allows users to quickly iterate through a design process driven by the interaction, emphasizing the action-perception loop as the essential component of the design process.

3.3 Architecture and Desirable Properties

This thesis is primarily concerned with the technical issue of learning a consistent mapping from user demonstrations. In the previous section we introduced the basic workflow from a user's perspective. Here, we describe the architecture and desirable properties of a MbD system.

3.3.1 Architecture

Figure 3.2 illustrates a proposition of architecture for a complete system for MbD.

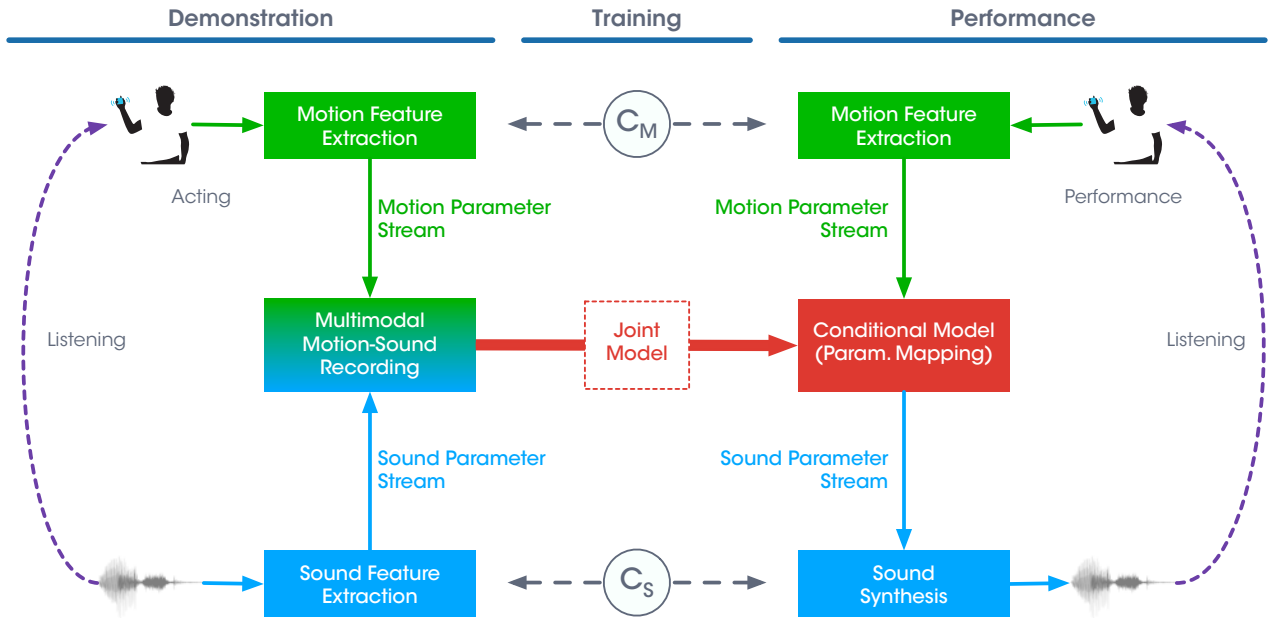


Figure 3.2: Architecture of a Mapping-by-Demonstration System. Blue and Green arrows respectively represent sound and movement, and the gray dashed arrows illustrate the correspondences.

The demonstrations are built by jointly and synchronously recording the motion and sound parameter streams originating from motion sensing and feature extraction, and sound feature extraction components. We exploit these multimodal sequences to train a model of the mapping that encode the relationship between motion and sound parameter sequences. We call *parameter mapping* this conditional model that expresses the dependencies of the sound control parameters over the input motion.

In performance, movements are described using the same feature extraction, and the parameters are streamed to the mapping layer that continuously generates the associated parameters. The generated sound parameters can finally be streamed to a sound synthesis engine. The figure depicts as C_M and C_S the motion and sound correspondences, that are further defined in Section 3.4.

Note: Thereafter, we denote by *mapping* — or *parameter mapping* — the layer linking motion parameters and sound parameters that is learned by a machine learning model. However, in the term Mapping-by-Demonstration (MbD) we consider the more general meaning of the word *mapping* as the relationship between the physical movement of the performer and the sound output.

3.3.2 Requirements for Interactive Machine Learning

We now derive a set of requirements of the mapping layer that are necessary to implement a *fluid* interaction design workflow (Zamborlin et al., 2014).

First, the training must be quick, as users might often adjust the parameters, evaluate the results through direct interaction, and quickly iterate in the design of the model.

Second, the model must be able to learn from few examples. The approach aims to give users the ability to intuitively design for their own idiosyncrasies: all the training examples are provided by the user through direct demonstration, which prevents the use of larger databases.

The third and most critical desirable property is relevance and consistency. Ideally, the model should be able to identify or learn what features and properties of the demonstrated relationship are relevant to the user. Or, alternatively, the models should have transparent properties, so users can easily understand and adjust its parameters.

Note that the transparency of the training process might depend on the expertise of the user, and, by extension, to the context of use. In specific settings, such as public installations, the training phase might not be revealed to novice users.

3.3.3 Properties of the Parameter Mapping

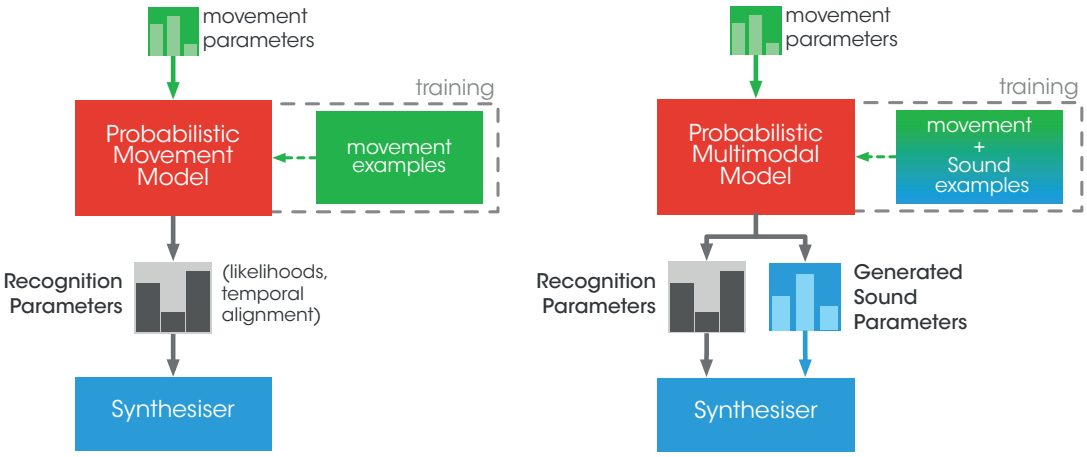
We now propose two criteria defining the properties of the parameter mapping layer, that originate from the strategies identified in the mapping through listening approach. We formalized in Caramiaux et al. (2014c) three types

of mapping: *instantaneous*, *temporal*, and *metaphoric* (see Section 2.4.2). These categories differentiate various levels of congruence between motion and sound parameters, that we propose to formalize with two modeling criteria: multimodality, and temporal modeling.

MULTIMODALITY AND PARAMETER GENERATION

We make a distinction between movement models and multimodal models that encode the motion-sound mapping through regression. Many approaches to sound control involving gesture recognition are based on movement models that are not intrinsically related to sound modeling (Figure 3.3a). In this case, the user defines the mapping between the recognition parameters and the input parameters of a synthesizer — possibly defining several gesture and sound classes. Such mappings could consist, for example, in triggering a particular sound segment each time a particular gesture is recognized. More advanced mappings may allow for aligning the playback of a sound segment to the performance of a gesture (Bevilacqua et al., 2011).

Alternatively, multimodal models are trained with sequences of joint movement-sound representations and therefore enables to learn movement-sound relationships using regression methods (Figure 3.3b). Consequently, these probabilistic models allow for directly generating sound features — or synthesis parameters — from motion features input into a trained system.



(a) Movement Model (b) Multimodal Model

Figure 3.3: Probabilistic sound control strategies based on movement models and multimodal models.

INSTANTANEOUS AND TEMPORAL MODELS

We differentiate *instantaneous* models from *temporal* models. Instantaneous models learn and perform static instantaneous movement-sound relationships without taking into account any temporal modeling. Practically, this means that the recognition or generation performed by the model at any given instant is independent of previous input. On the contrary, *temporal* models take into account time series. In this case, the recognition or generation performed by the model depends on the history of the input. The choice of

the level of temporal modeling might depend on the application and context of use of the interactive system.

3.3.4 Proposed Models

In this dissertation, we mostly focus on the parameter mapping layer that learns the interaction model between movement and sound parameter sequences. Precisely, we propose a set of models addressing all combinations of the multimodality and temporal modeling criteria that we identified as crucial for modeling motion-sound relationships.

The models are summarized in Table 3.1. We implemented two instantaneous models based on Gaussian Mixture Models and two temporal models with a hierarchical structure, based on an extension of the basic Hidden Markov Model (HMM) formalism. The movement models and the multimodal models are described and analyzed in Chapters 4 and 6, respectively.

	Movement	Multimodal
Instantaneous	Gaussian Mixture Model **GMM**	Gaussian Mixture Regression **GMR**
Temporal	(Hierarchical) Hidden Markov Model **(H)HMM**	(Hierarchical) Hidden Markov Regression **(H)HMR**

Table 3.1: Summary of the proposed probabilistic models

3.4 The notion of Correspondence

We propose to define the notion of correspondence to characterize the match between the motion and sound representations used in demonstration and those used in performance.

Each research field studying social learning, imitation or mimicry deals with the idea of correspondence (Dautenhahn and Nehaniv, 2002). Argall et al. (2009) propose the following definition for the case of robot Programming-by-Demonstration:

> The issue of correspondence deals with the identification of a mapping between the teacher and the learner that allows the transfer of information from one to the other.

We reviewed in Section 2.5.3 the necessity for additional mappings in robot motor learning, between the recorded execution and the respective movements of the teacher and the learner.

A similar issue can be identified in the case of motion-sound Mapping-by-Demonstration, depending on the coherence between the modalities

used in *demonstration* and *performance*. In many cases, the type of sounds used in demonstration and performance might not match accurately. For example, the demonstration set might only contain a small portion of the palette available in performance, or the modality can even be different (e.g. if the demonstrations are vocal imitations of the performance sounds). For these reasons, non-trivial correspondence can occur, that is therefore conditioned on the processes of feature extraction and sound synthesis.

MOVEMENT CORRESPONDENCE

The *movement correspondence* C_M defines the match between the sensors and motion features used for capturing and describing movement in the demonstration and performance phases (see Figure 3.2, top).

While it can be surprising to define a non-trivial correspondence for the movement representation in demonstration and performance, we aim to provide the most general framework. We can imagine several applications where the movement representation differs. For example, one could use elaborate motion capture techniques to learn expert behaviors, and transfer the learned associations to a cheaper sensing system for public demonstration — i.e. from full-body marker-based motion capture to a Kinect. At a higher-level, we can imagine 'sketching' motion-sound associations in one modality, before transferring to a full-body situation.

For simplicity, in this work we only consider the case where the same sensors and movement features are used in both phases. Therefore, we assume the property that the movement correspondence is identity:

$$C_G^{(M)} = \boldsymbol{I}_d \tag{3.1}$$

SOUND CORRESPONDENCE

The *sound correspondence* C_S represents the match between the sounds used as demonstration and those generated by the synthesis engine in performance (see Figure 3.2, bottom).

Correspondence on the sound modality is more challenging, as it relates to the problem of consistency between sound analysis and synthesis. The sound correspondence will equal identity when the demonstration and performance sounds belong to the same class, and when the analysis/synthesis framework is ideal — i.e. when the feature extraction is a perfect model inversion of the synthesis engine.

We can define several levels of correspondence to analyze how closely the example sounds match the synthesized sounds. In the following, F refers to the sound feature extraction used to extract sound parameters during demonstration, and G refers to the sound synthesis that translate sound parameters to the acoustic waveform.

A *strong correspondence* occurs when the demonstration sounds are synthesized using the same synthesis engine as used for performance. In this case we have a direct access to the sequence of exact parameters associated to a sound example. Virtually, the feature extraction corresponds to an ideal inversion of the synthesis model:

STRONG CORRESPONDENCE: $C_S = \boldsymbol{I}_d \quad \Leftrightarrow \quad F = G^{-1}$

We now consider that the sound examples originate from audio recordings. In this case, the sound parameters are computed using either model inversion, or using audio feature extraction. As the sound analysis/synthesis framework is likely to introduce artifacts in sound resynthesis, this situation yields a *weak correspondence*:

WEAK CORRESPONDENCE: $C_S \approx \boldsymbol{I}_d \quad \Leftrightarrow \quad F \approx G^{-1}$

Finally, a *mismatch* can occur when different types of sounds are used for demonstration and performance. This is for example the case when the demonstration consists of vocal imitations, whereas the sound used for synthesis belong to another corpus. This situation involves important remapping operations to associate the different types of sounds, yielding a non trivial correspondence:

MISMATCH: $C_S \neq \boldsymbol{I}_d \quad \Leftrightarrow \quad F \neq G^{-1}$

Along this thesis, we present several systems with varying degrees of correspondence. We present in Section 6.3 an application using physical modeling sound synthesis, where the same synthesis engine is used for generating the example and performance sounds. The systems presented in Chapters 8 and 9 use recorded sound, and have therefore a weaker match.

3.5 Summary and Contributions

This chapter formalized the concept of Mapping-by-Demonstration (MbD) as a framework 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 proposed to learn the mapping between motion and sound parameters from multimodal demonstrations using probabilistic models. We analyzed the requirements of the computational models of the mapping and we derived two criteria for analyzing the models: multimodality and temporal modeling. Our analysis led to the definition of the notion of *correspondence* between movement (resp. sound) modalities in demonstration and performance, that might vary with the application.

The four following chapters describe and evaluate the different probabilistic models. The theory for probabilistic movement models and probabilistic models for sound parameter generation is presented in

Chapters 4 and 6, respectively. Chapters 5 and 7 present an application and evaluation of both types of models to movement analysis, in the case of Tai Chi performance.

4

Probabilistic Movement Models

We proposed in Section 3.3.3 to differentiate probabilistic models of movement from joint models of the motion-sound relationships. This chapter investigates the former category, as a first iteration in the development of the Mapping-by-Demonstration (MbD) framework. We discuss the formalism and applications of three probabilistic movement models with varying levels of temporal modeling: Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and Hierarchical Hidden Markov Models (HHMMs). After presenting the representation, training and inference of each model, we detail two main contributions. First, the originality of our approach lies in the interactive machine learning implementation of the probabilistic models that relies on user-defined parameters of regularization of complexity. Second, we present a set of applications of movement models to Mapping-by-Demonstration, and we formalize several sonic interaction design patterns using continuous recognition.

OUTLINE We start this chapter by presenting two well-known probabilistic models, Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), that are formally described in Sections 4.1 and 4.3. This allows us to present the simplest cases of MbD that draw upon continuous gesture recognition and following (Sections 4.2 and 4.4). We detail these models to highlight key features of our approach in interactive machine learning. First, explicitly controlling the regularization allows users to adapt the models to gesture variations learned from few examples. Second, the use of the forward inference in HMMs guarantees low-latency recognition and mapping in performance. We then step up complexity by presenting our implementation of the Hierarchical Hidden Markov Model (HHMM) in Section 4.5. The HHMM enriches the temporal structure of the sound synthesis, making possible the development of sound control strategies using segment-level mapping (Section 4.6).

NOTATION We now define the mathematical conventions. In the following, we denote vectors and matrices by bold letters $\boldsymbol{x}$, $\boldsymbol{A}$. An ensemble of values is denoted using brackets, for example $\{1 \cdots N\}$, and we use the subscript notation $\boldsymbol{x}_{t1:t2} = \{x_{t1}, \cdots, \boldsymbol{x}_{t2}\}$ for data segments. Proba-

bility density functions are noted $p(\bullet)$ while their conditional counterparts are noted $p(\bullet \mid \bullet)$. By convention, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are exclusively used to designate the mean vector and covariance matrix of Gaussian distributions $\mathcal{N}\left(\bullet; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$. In latent variable models, z refer to hidden states while the observed variables are noted $\boldsymbol{x}$.

4.1 Movement Modeling using Gaussian Mixture Models

Gaussian Mixture Model (GMM) — also called Mixture of Gaussian (MOG) — is one of the most widespread model of the family of finite mixture models (McLachlan and Peel, 2004). In a mixture model, any density can be approximated by a finite weighted sum of base distributions; in the case of GMMs, multivariate Gaussian distributions. The model assumes the independence of successive observations, and therefore does not account for a representation of movements as time processes. In practice, this assumption implies that the model is static: when performing recognition, the results at a given time step are independent from previous observations. The model evaluates the likelihood of the input data on a frame-by-frame basis, and therefore belongs to the family of *instantaneous* models we defined in Section 3.3.3.

In this section, we outline the representation, learning and recognition methods for GMMs. We complement this presentation with the specificities of our user-centered implementation.

4.1.1 Representation

Within our applicative framework, we use GMMs to encode movements represented in continuous parameter spaces. We consider a set $\{\mathscr{X}\} = \{\boldsymbol{x}_i\}_{i=1:T}$ of observations from recordings of movement performances, each represented by a sequence of frames $\boldsymbol{x}_i \in \mathbb{R}^D$ sampled from a D-dimensional stream of movement features. In a GMM, this dataset is modeled by a mixture of K Gaussian components, defined by the probability density function (pdf)

$$p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{4.1}$$

The model is described by a set of parameters $\boldsymbol{\theta} = \{w_{1\cdots K}, \boldsymbol{\mu}_{1\cdots K}, \boldsymbol{\Sigma}_{1\cdots K}\}$ where

w_k	is the prior probability (or weight) of the kth component $w_k \geq 0$, $\sum_{k=1}^{K} w_k = 1$
$\boldsymbol{\mu}_k \in \mathbb{R}^D$	is the mean vector of the kth component
$\boldsymbol{\Sigma}_k$	is the $D \times D$ covariance matrix of the kth component

As a reminder, the pdf of a Gaussian distribution of mean $\boldsymbol{\mu}$ and a covariance $\boldsymbol{\Sigma}$ is defined by

$$\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{4.2}$$

The model therefore approximates an ensemble of training data through a weighted sum of Gaussian distributions, as illustrated in Figure 4.1.

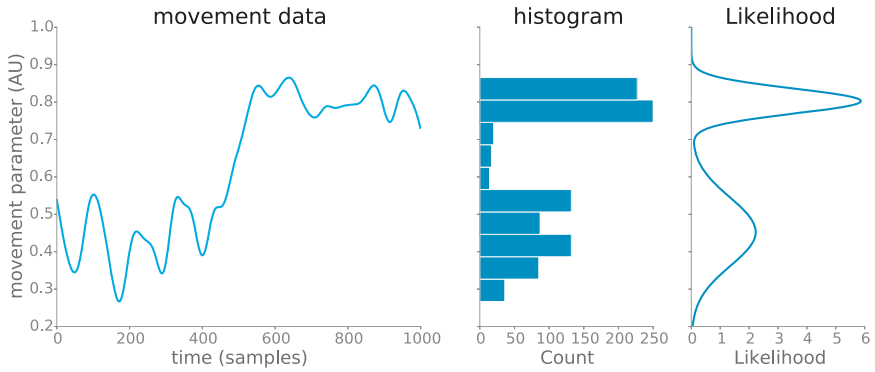


Figure 4.1: Movement Modeling with GMMs. The movement data was synthesized from filtered noise, and the gmm was trained with parameters $\{K = 2, \lfloor\sigma\rfloor = [1e^{-3}, 1e^{-3}]\}$.

4.1.2 Learning

EXPECTATION-MAXIMIZATION ALGORITHM

The parameters of a GMM can be learned through the Expectation-Maximization (EM) algorithm. The EM estimates the parameters of the model (the weight, mean and covariance of each component) that maximize the likelihood of the training data. The method iteratively estimates the parameters of the model through two steps of expectation and maximization, that guarantee the increase of the log-likelihood. For a complete derivation of the algorithm, see Bilmes (1998) or Murphy (2012, Chapter 11).

CONVERGENCE CRITERION

The update equations of the EM algorithm guarantee that the likelihood increases at each iteration, ensuring the convergence to a local maximum. The algorithm has converged when the log-likelihood stops changing. In practice, this criterion might be difficult to reach in some cases, and users might want to constrain training to ensure quick convergence. In our implementation, users can define two possible convergence criteria:

MAXIMUM NUMBER OF STEPS is the fixed number of EM iterations to perform to ensure convergence.

RELATIVE LOG-LIKELIHOOD PERCENT-CHANGE states that the algorithm has converged when the percent of change of the relative log likelihood is inferior to a given threshold.

The iterative estimation of the parameters using the EM algorithm is illustrated in Figure 4.2 where a GMM with 2 components is trained on synthetic data.

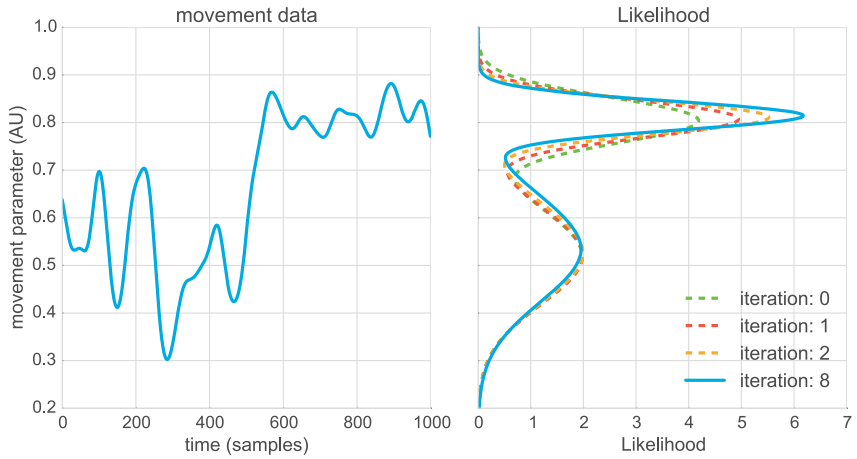


Figure 4.2: Training a GMM with the EM algorithm. The optimal estimate was reached after 8 iterations. The GMM was trained with parameters $\{K = 2, \lfloor\sigma\rfloor = [1e^{-5}, 1e^{-5}]\}$.

INITIALIZATION The GMM has been implemented with respect to the interactive machine learning workflow presented in Chapter 3. As stressed earlier, one of the main constraints of sonic interaction design is the number of available training examples. Therefore, the initial estimation of the model parameters for the EM algorithm is crucial. In our implementation, we chose two strategies for pre-estimating the models parameters according to the size of the training set:

FULLY OBSERVED APPROXIMATION If the training set contains a single phrase, we distribute the Gaussian components along the training example, which gives a first segmentation of the example. This method is sufficient in most of the cases, especially when the training data has a low redundancy.

BIASED K-MEANS If the training set contains several phrases, a K-Means algorithm is used to determine the initial position of the centroids within the training data. This K-Means is initialized using the first phrase of the training set — hence the name *biased*, — that proved to converge more quickly and consistently that random initialization.

4.1.3 Number of Components and Model Selection

NUMBER OF COMPONENTS The number of Gaussian Components (K) defines the complexity — or non-linearity — of the model. Choosing the appropriate number of components for a specific application, e.g. classification, can be a difficult task.

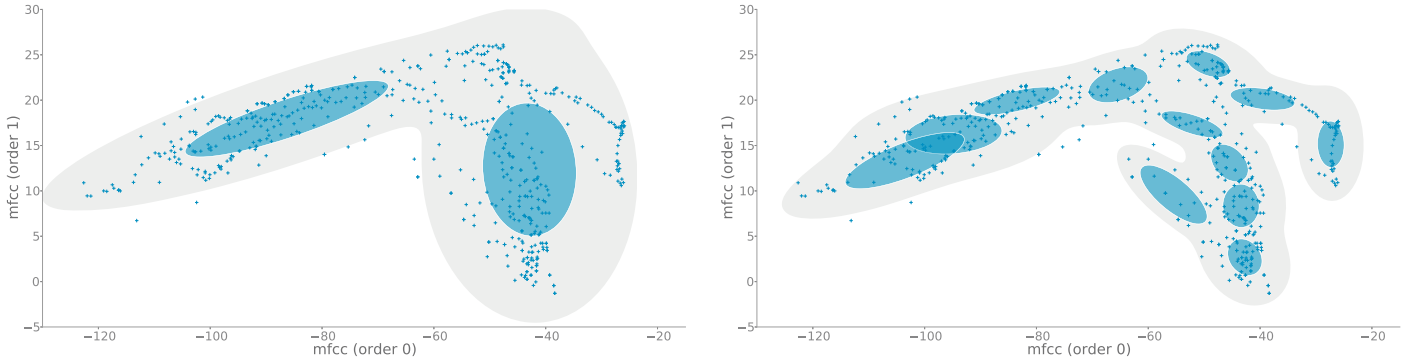
A small number of components can result in a simple model that will be less discriminative. On the other hand, increasing the number of Gaussians is likely to result in overfitting, therefore losing the generalization of the model to new observations.

MODEL SELECTION Several methods have been proposed in the machine learning literature for automatically selecting the optimal number of components. For example, one can perform the training with several values of the number of components, and then select the model optimizing a given criterion (e.g. cross-validation, Bayesian Information Criterion).

However, these methods might not be of critical interest in Mapping-by-Demonstration. First, they require to perform the training multiple times with various parameters to find the optimal value, which increases the training time. Second, the optimization criterion might not be relevant from the user's viewpoint. Fiebrink et al. (2011) highlights that composers systematically prefer direct evaluation to cross-validation when building classifiers.

In this work, we do not investigate further methods for automatic selection of the models parameters. We argue that direct evaluation is a very efficient way to let users optimize the model themselves. Our implementation focuses on quick training and on a short interaction loop that allows users to rapidly evaluate different alternatives.

The influence of the number of Gaussians is illustrated in Figure 4.3 that depicts the 95% confidence interval (CI) ellipse of the Gaussian components. A GMM is trained on several recordings of the vowel 'A' performed by professional singer Marie Soubestre. These examples were recorded during the production of Januibe Tejera's Cursus piece "Le patois du Monarque".



(a) Fitting with 2 Gaussians

(b) Fitting with 12 Gaussians

Figure 4.3: Influence of the number of components in Gaussian Mixture Models. The data consists of the MFCCs extracted from several performances of single vowels by a singer. Performed by Marie Soubestre, from Januibe Tejera's "Le patois du monarque"

4.1.4 User-adaptable Regularization

We propose to use regularization to deal with the issue of learning from small training sets. Our implementation makes regularization explicit to users. We implemented regularization through a prior $\lfloor\sigma\rfloor$ added to the covariance matrices of the Gaussian distributions at each re-estimation in the

EM algorithm. Our regularization method is a special case of the Bayesian regularization technique proposed by Ormoneit and Tresp (1996), and can be viewed as a special case of the MAP estimation detailed in Murphy (2012, Chapter 11).

The goal of this parameter is twofold: it prevents numerical errors during training by avoiding that variances tend towards zero, and it 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.

$\lfloor\sigma\rfloor$ combines a relative prior and an absolute prior:

- $\lfloor\sigma\rfloor^{(rel)}$ (*Relative Regularization*) is proportional to the variance of the training set $\mathscr{X}$ on each dimension
- $\lfloor\sigma\rfloor^{(abs)}$ (*Absolute Regularization*) represents the absolute minimal value to add to the diagonal.

At each iteration of the EM algorithm, we estimate the regularized covariance matrix $\bar{\mathbf{\Sigma}}$ from the covariance matrix $\mathbf{\Sigma}$ estimated via EM as

$$\bar{\mathbf{\Sigma}} = \mathbf{\Sigma} + \max\left(\lfloor\sigma\rfloor^{(rel)} * Var(\mathscr{X}), \lfloor\sigma\rfloor^{(abs)} * \mathbf{1}_D\right) \cdot \boldsymbol{I}_D \tag{4.3}$$

4.2 Designing Sonic Interactions with GMMs

In this section, we propose a simple application using Gaussian Mixture Models (GMMs) for the continuous recognition of scratching modes. It is based on the association between different 'qualities' of surface gestures and resonant models.

GMMs are a versatile model for sonic interaction design. While their use is often restricted to simple classification tasks, we argue that their semi-parametric approach to density estimation offers a wider range of strategies for designing interactive systems.

We start by describing the *scratching* application, which leads us to discuss classification, continuous recognition, and spotting. We also discuss how regularization can be used for generalizing from few examples.

4.2.1 The *Scratching* Application

We consider a proof-of-concept application using surface gestures as an input method for sound control. We focus on the 'quality' of surface gestures rather than on their trajectory, approaching the movement representation as the way the surface is touched. We capture gestures using a single contact microphone. The touch quality of the gesture is embedded in the timbre of the audio, and we propose to use Mel-Frequency Cepstral Coefficients (MFCCs) as features to represent the movement. This representation allows us to discriminate several 'scratching modes' — such as *rubbing*, *scratching* or *tapping* — using GMMs-based continuous recognition, as described in Françoise et al. (2014a). The application represents a first itera-

tion in the development of the Mapping-by-Demonstration (MbD) methodology, and can be viewed as one of its most basic instance.

This application builds upon research in the *{Sound Music Movement} Interaction* team at Ircam. In particular, GMM-based timbre recognition was initially prototyped by Rasamimanana and Bloit (2011). The approach was later extended by Zamborlin (2015) who developed the *Mogees*[1] system.

The interaction loop of the system is as follows. In *Demonstration*, the user can record several surface gestures. The audio from the microphone is recorded and analyzed to extract MFCCs. The user then annotates the training set by associating a resonant filter model to each recorded scratching mode, and we train a GMM for each class. The flowchart of the system in the *performance* phase is illustrated in Figure 4.4. The surface gesture is captured with the contact microphone and the audio is streamed to each of the resonant models. We use the GMMs for continuous recognition: the posterior likelihood of each class defines the gain applied to the output of the associated resonant model. A demonstration video can be found online.[2]

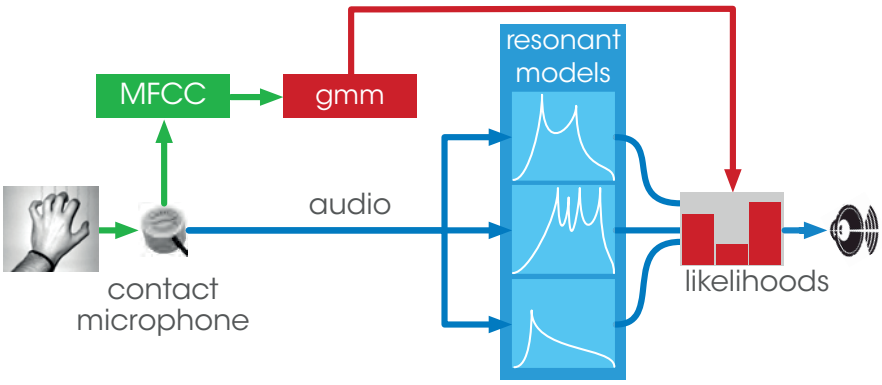


Figure 4.4: *Scratching*: GMM-based Surface gesture recognition and mapping. The audio signal from a contact microphone is used to recognize different scratching modes. Each playing mode is associated to a resonant filter used to process the input audio stream. The posterior likelihoods continuously control the intensity of each filter.

The subsequent sections discuss how GMMs can be used for classification, continuous recognition, and spotting.

4.2.2 From Classification to Continuous Recognition

In Section 4.1, we described the formalism and algorithms of GMMs for movement modeling. Here, we address the supervised learning problem of recognition and classification. We consider a set of C classes to recognize. One model is trained with the EM algorithm for each class $c \in \{1 \cdots C\}$ using a subset $\{\mathscr{X}_c\}$ of training examples. The training data must be therefore be labeled. We can then express the likelihood functions as class-conditional

1 http://mogees.co.uk/
2 http://vimeo.com/julesfrancoise/nime2014

likelihoods $p(\boldsymbol{x} \mid c) = p(\boldsymbol{x} \mid \theta_c)$. The behavior of a GMM in performance in illustrated in Figure 4.5: the model associated to each class evaluates the likelihood of the movement features of the current frame.

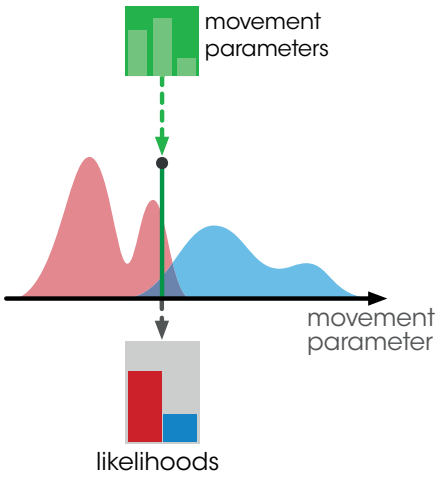


Figure 4.5: Recognition with GMMs.

CLASSIFICATION Deriving the maximum likelihood classifier is therefore straightforward. Using Bayes' rule, we can write the posterior probability of class c as

$$p(c \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid c)}{\sum_{c'=1}^{C} p(\boldsymbol{x}|c')} \tag{4.4}$$

provided that we assume a equal prior on each class ($p(c) = 1/C$). Therefore, the classification can be performed by selecting the class maximizing the posterior probability.

CONTINUOUS RECOGNITION The probabilistic nature of the model offers several advantages over simple classification. Indeed, when performing inference, we continuously estimate the likelihood of each class. This quantity defines a confidence measure that can be used as a continuous control parameter for interacting with sound synthesis. While classification allows for discrete interaction paradigms such as triggering or selection, using the likelihoods as continuous control offers richer interaction techniques. It is often advantageous to use the posterior class probabilities, as defined in Equation 4.4, that are normalized across classes.

EXAMPLE As an example, we consider recordings from the *scratching* application. Figure 4.6 depicts three single-Gaussian GMMs trained on MFCC data originating from a laptop microphone. Each class is defined by a specific scratching mode: rubbing (blue), scratching (red), tapping (yellow). Each model was trained using a performance of each scratching mode (about 10 seconds long). At the bottom of the figure are depicted the

training datapoints, and their associated model is represented by the 95% CI ellipse of the Gaussians.[3]

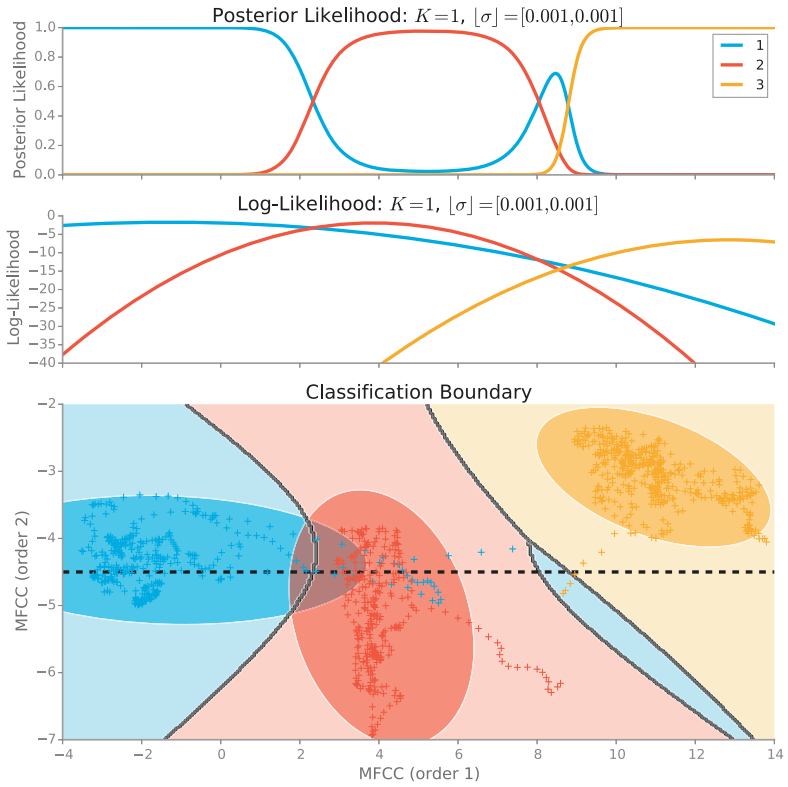


Figure 4.6: Classification and Continuous Recognition with GMMs. 3 single-Gaussian GMMs are trained on MFCC data originating from a microphone. Each class represents a distinct scratching mode: *rubbing* (blue), *scratching* (red), *tapping* (yellow). On the bottom graph, the ellipses represent the 95% CI of each Gaussian, the colored region correspond to the classification boundaries, and the training data is represented by point clouds. The top graph represents the posterior likelihood of each class along the black line ($y = -4.5$), while the middle graph draws the log-likelihoods.

The classification boundaries are represented by the colored areas that defines the regions of the input data where each class maximizes the posterior probability. The top curves represent respectively the posterior likelihood and the log-likelihood of each class, for a test data vector represented by the black dashed line ($y = -4.5$). These quantities evolve continuously, interpolating between the various models, and can therefore be used for continuous sound control. It is interesting to note that posteriors and log-likelihoods provide different representations of the uncertainty, one that is normalized, not the other.

3 For details on the computation of 95% confidence interval ellipses, see Murphy (2006)

SPOTTING Gesture *spotting* (Yang et al., 2007; Kang, 2004; Kim et al., 2007) refers to the segmentation problem of discerning "meaningful" gestures in a continuous stream of movement. It aims at discarding movements that are not meant to be tracked or utilized for sound control, such as ancillary movements in the case of music performance. In GMMs, we can perform posture or gesture spotting using two strategies:

LOG-LIKELIHOOD THRESHOLD The simplest solution for spotting specific poses is to use a fixed threshold on the log-likelihood. Segmentation can therefore be performed by defining segments $\boldsymbol{x}_{a:b}$such as$\forall i \in \{a \cdots b\}\ \log p(\boldsymbol{x}_i|,\theta) > \log p_{thresh,\theta}$

FILLER MODEL An alternative consists in defining one or several *filler* model trained on all the training data that is not labeled as a meaningful gesture (Eickeler et al., 1998). This model is considered as a class that runs competitively against the other classes.

Each spotting technique has advantages and shortcomings. Fixed thresholding is both simple and easier to use as it doesn't require additional training data. However, it might be difficult to specify the threshold manually: it can vary between gestures, and it depends on the model's parameters (number of components, regularization). On the other hand, using a filler model brings more flexibility in that the thresholds are defined by the competition between models.

We give a brief overview of the problem of spotting in Section 5.4.4, where we compare instantaneous and temporal models for gesture spotting.

CLUSTERING GMMs can also be used for unsupervised learning problems such as clustering. GMMs model arbitrary densities through a mixture of basis distributions. Clustering can thus be performed by utilizing the mean and covariance of each component as the center and width of a cluster to partition the input space. We can perform clustering using a GMM θ with K components using the posterior likelihood of each component

$$p(k \mid \boldsymbol{x}) = \frac{w_k \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{k'=1}^{K} w_{k'} \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}\right)} \tag{4.5}$$

These quantities can either be used for discrete clustering — choosing the component that maximizes the posterior, — or as additional continuous control parameters, bringing a finer level of detail.

Unsupervised learning is not the primary focus on this thesis. However, clustering might be a good initial approach to discover patterns within data. For example, one could use clustering to identify a set of postures that are clearly identified by the model, as a first step towards the creation of a system where these poses are supervisedly associated to sound parameters or classes.

In the *scratching* example, one could use clustering to identify possible scratching modes. For this purpose, we can train GMM with a large number of Gaussians over a recording containing all kinds of surface interactions. Examining the posterior likelihoods of each component during the

performance phase would indicate the playing modes that are effectively identified by the system, therefore informing a possible supervised recognizer in a subsequent design iteration.

4.2.3 User-Defined Regularization

We introduced a regularization parameter in Section 4.1.4, that lets users specify a prior added to the variance of the Gaussian components of the mixture. This prior ensures the convergence on small training sets, and is useful to avoid numerical errors. In this section, we show how this parameter can be used for continuous control.

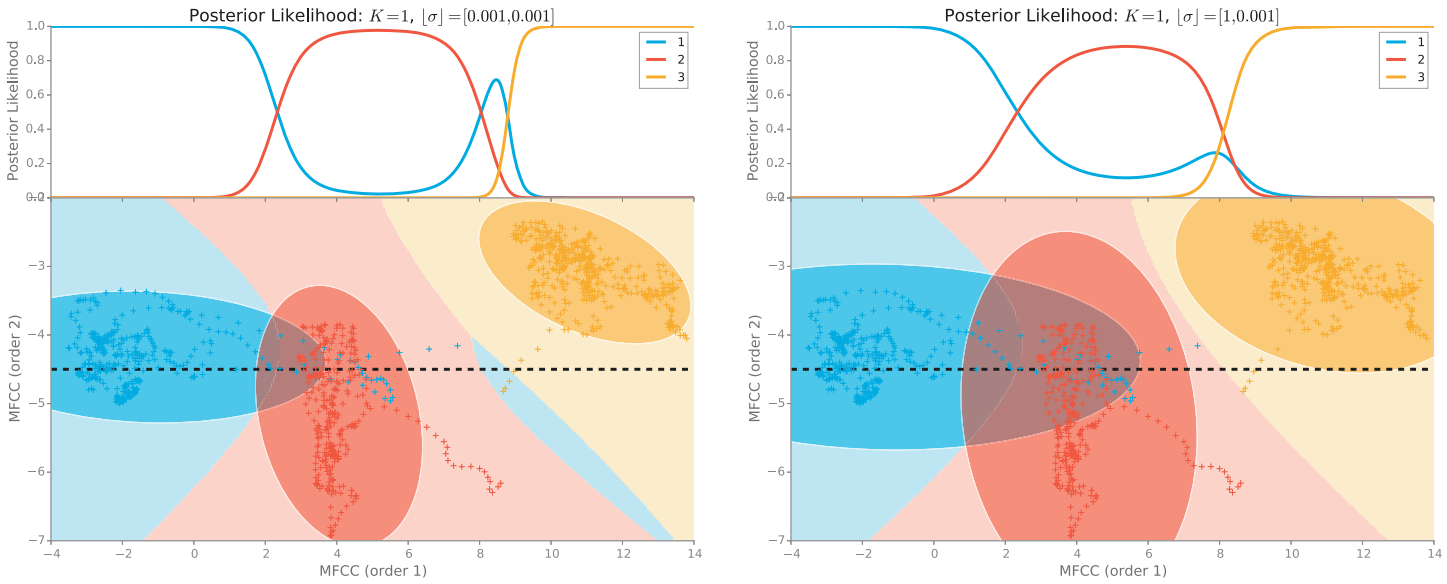
We argue that regularization impacts on the smoothing of the recognition without necessarily affecting the classification boundary. Indeed, artificially increasing the variance naturally increases the overlap between several Gaussian distributions, and therefore can smooth the transition between several models.

Figure 4.7 illustrates the use of regularization for the example of the *scratching* application presented above. The figure depicts the models learned for two values of the regularization. We can observe that regularization increases the overlap between the models, and provides a smoother transition of the posterior likelihoods between classes.

4.2.4 Discussion

We presented a simple example of MbD application using Gaussian Mixture Models (GMMs) for movement modeling. The application allows users to associate qualities of surface gestures to resonant models. We utilized this application to formalize and discuss a set of design patterns based on GMMs.

We showed that beyond the simple classification of postures of gestures, the model allows for continuous control through the evaluation of the likelihoods. We specified several parameters for improving the usability of the recognition in interactive contexts. In particular, regularization can be used to adjust the behavior of the recognizers, allowing various degrees of accuracy, responsiveness, and stability. The interaction design patterns based on GMMs are summarized in Figure 4.8.



(a) Small regularization

(b) Large regularization

Figure 4.7: Regularization for GMM Recognition for the scratching application. 3 single-Gaussian GMMs are trained on MFCC data originating from a microphone. Each class represents a distinct scratching mode: *rubbing* (blue), *scratching* (red), *tapping* (yellow). On the bottom graph, the ellipses represent the 95% CI of each Gaussian, the colored region correspond to the classification boundaries, and the training data is represented by point clouds. The top graph represents the posterior likelihood of each class along the black dashed line ($y=-4.5$).

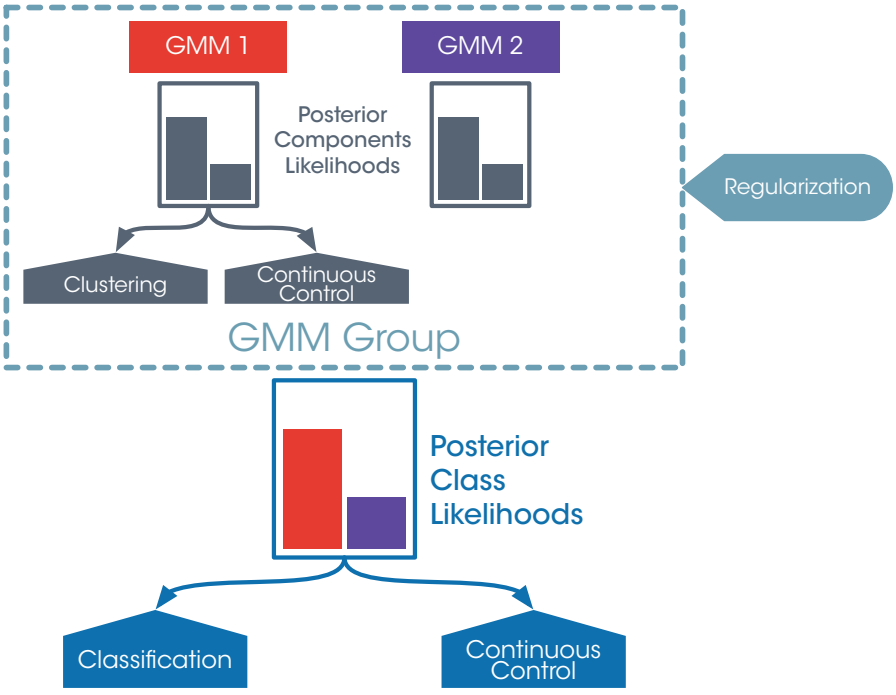


Figure 4.8: Summary of the Design Strategies based on Gaussian Mixture Models.

4.3 Movement Modeling using Hidden Markov Models

In the previous sections, we introduced Gaussian Mixture Models (GMMs), an efficient statistical model for estimating densities with arbitrary shape and potentially high-dimension. Unfortunately, assuming the independence of observations can be extremely limiting for modeling sequential data, where both short-term and long-term dependencies play a crucial role. In particular, dynamic gesture modeling requires taking into account the temporal structure of the motion parameters over time.

To overcome the independence assumption of GMMs, we consider HMMs that have proved effective for modeling sequential data in a variety of domains such as Automatic Speech Recognition (Rabiner, 1989), speech synthesis (Tokuda et al., 2013), gesture recognition (Lee and Kim, 1999; Mitra, 2007; Bevilacqua et al., 2010), movement generation (Calinon et al., 2011; Tilmanne, 2013).

A HMM is a statistical model for time series analysis. It assumes that the observed data (or signal) is a noisy measurement of a system that can be modeled as a Markov process. It can be seen as a density model on sequences, that extends a mixture model through a first-order Markov dependency between latent variables. A HMM articulates a hidden discrete-time discrete-state Markov chain with an observation model.

Many tutorials address the formalism, algorithms, and limitations of HMMs: Rabiner (1989) is often cited as a reference[4], while Bilmes (2006) presents a thorough mathematical analysis of HMM's properties, and Murphy (2012, Chapter 12) gives a very clear description of the model and its extensions. In this section, we briefly examine the representation, learning, and inference algorithms for HMMs.

4.3.1 Representation

Consider a movement recorded as a sequence of observations $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t \in \mathbb{R}^D$ is a D-dimensional vector – that is, a frame extracted from a stream of movement parameters originating from sensors or feature extractors. The joint distribution of a HMM can be written using the hidden states $z_t \in \{1 \cdots N\}$ as

$$\begin{aligned} p(\boldsymbol{x}_{1:T}, z_{1:T}) &= p(z_{1:T})\, p(\boldsymbol{x}_{1:T} | z_{1:T}) \\ &= \underbrace{\left[p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1}) \right]}_{\text{(a) Markov process}} \underbrace{\left[\prod_{t=1}^{T} p(\boldsymbol{x}_t | z_t) \right]}_{\text{(b) observation model}} \end{aligned} \tag{4.6}$$

The first part *(a)* of equation 4.6 embodies the first-order Markov properties that asserts that the state at time t only depends on the state at $t-1$. The second part *(b)* represents the observation model that defines the state-

4 See also Rahimi's erratum for Rabiner's article (2000), available online.

conditional observation density distribution. These dependencies are illustrated in the Dynamic Bayesian Network representation of Figure 4.9.

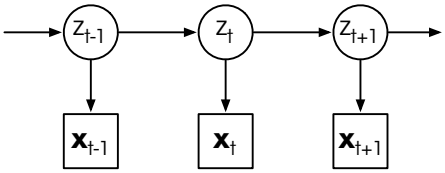


Figure 4.9: Dynamic Bayesian Network representation of a HMM. Horizontal arrows represent the first-order Markov process. Squares represent observed variables, round nodes represent the hidden states

In the case of discrete observations, the observation model can be defined as a matrix. In our case, movement being captured as a continuous process, we choose an continuous observation model; namely, a Gaussian distribution.[5] A N-state HMM is therefore defined by a set of parameters $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ constituted of a prior vector $\boldsymbol{\pi} = \{\pi_i\}$, a state transition matrix $\boldsymbol{A} = \{a_{ij}\}$, and an observation probability distribution $\boldsymbol{B} = \{b_j(\boldsymbol{x}_t)\}$ where

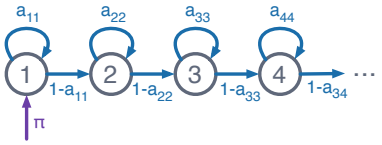
$$\begin{aligned}
\pi_i &\triangleq p(z_1 = i) && \text{is the prior probability of the } i\text{th state.} \\
&&& \pi_i \geq 0 \text{ and } \textstyle\sum_{i=1}^{N} \pi_i = 1 \\
a_{ij} &\triangleq p(z_t = j | z_{t-1} = i) && \text{is the probability of transiting from state } i \text{ to state } j. \\
&&& a_{ij} \geq 0 \text{ and } \textstyle\sum_{i=1}^{N} a_{ij} = 1 \\
b_j(\boldsymbol{x}_t) &\triangleq p(\boldsymbol{x}_t \mid z_t = j) && \text{is the observation probability distribution.} \\
&= \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) && b_j(\boldsymbol{x}_t) \geq 0 \text{ and } \textstyle\int_{\boldsymbol{x}_t} b_j(\boldsymbol{x}_t) d\boldsymbol{x}_t = 1
\end{aligned}$$

TOPOLOGY Several topologies of the Markov chain can be specified through the transition matrix. Without prior knowledge a full transition matrix can be used; however, it is usual to choose a left-right topologies for modeling temporal processes. The transition probability matrix is then triangular, meaning that transitions can only be made in the direction of time, and respects the properties

$$a_{ij} = 0 \quad \forall j < i, \forall j > i + \Delta$$

where Δ represents the maximum number of states that can be skipped. Figure 4.10 gives an example of topology for a 4-state left-right HMM.

5 In practice, our implementation allows the use of Gaussian Mixture Model as a continuous distribution, therefore increasing the complexity of the observation model. In this case, the observation probability distribution for a mixture with K components is defined by $b_j(\boldsymbol{x}_t) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k}\right)$. However, as most of our application involve small-size training sets, we often limit to a single Gaussian distribution per state.



(a) Graphical model

$$\pi_i^{(m)} = \delta(i,1)$$
$$a_{ij}^{(m)} = a_{ii}\delta(i,j) + (1-a_{ii})\delta(i,j-1)$$

(b) Transition Parameters

Figure 4.10: Graphical model and transition matrix of a HMM with left-right topology

4.3.2 Inference

TYPES OF INFERENCE This section discusses the algorithms allowing to infer the state sequence associated to an observation sequence for a HMM with known parameters (Rabiner, 1989; Murphy, 2012). We differentiate several types of inference for sequence models:

FILTERING is an online (causal) estimation of the *belief state* $p(z_t|\boldsymbol{x}_{1:t})$ computed with the Forward algorithm.

SMOOTHING is an offline estimation of the belief state $p(z_t|\boldsymbol{x}_{1:T})$. Smoothing may decrease uncertainty about the belief state, but requires the entire observation sequence.

FIXED-LAG SMOOTHING can be an interesting compromise between the online and offline approaches. It consists in computing $p(z_{t-\tau}|\boldsymbol{x}_{1:T})$, therefore providing belief state estimation with a fixed delay τ, but gaining in certainty.

PREDICTION aims to predict the future given past observations; namely, computing $p(z_{t+\tau}|\boldsymbol{x}_{1:t})$ with an *horizon* $\tau > 0$.

MAXIMUM A POSTERIORI (MAP) ESTIMATION computes the most probable state sequence by evaluating $\text{argmax}_{z_{1:T}}\, p(z_{1:T}|\boldsymbol{x}_{1:T})$. It is solved using the Viterbi Algorithm.

Our applications focus on continuous interaction, that requires to perform inference in real-time, either causally or with very low latency. Therefore, we primarily use filtering, that estimates the state probabilities causally using the forward algorithm. We now discuss this choice with respect to the other types of inference.

MAP estimation is restricted to the set of cases where the entire sequence of observations is available. The Viterbi algorithm combines a forward pass — also known as the *max-product* algorithm, that causally estimate $\delta_t(j) = \max_{z_1,\cdots,z_{t-1}} p(z_t = j, z_{1:t-1}|x_{1:t})$, — and a backtracking operation that finds the optimal state sequence respecting the transition structure. While it could be tempting to use only the forward pass of the algorithm, this is not sufficient to guarantee the consistency of the state sequence. This makes the *max-product* algorithm less relevant and more prone to errors than the forward algorithm that sums over all possible state sequences.

Alternatively, Bloit and Rodet (2008) and Sramek (2007) proposed online implementations of the Viterbi algorithm that compute the optimal path according to the MAP criterion with low latency. However, these methods introduce a variable length delay which can be difficult to manage for continuous interaction. Recently, Ravet et al. (2014) proposed a sliding window

method and a state stability method. Both algorithms are approximations of the Viterbi decoding that improve online the classification accuracy, but introduce a delay in the recognition.

It is important to stress that while guaranteeing a consistent state sequence can be crucial in applications such as Automatic Speech Recognition (ASR) in order to respect linguistic constraints, it is less important for movement modeling. We consider movement as continuous time process; therefore, the transitions in the left-right model support the timing of the trajectory rather than they encode a set of transition 'rules'. MAP estimation relies on a choice of the best transition at each update, that is likely to propagate errors. On the contrary, the forward algorithm cumulates all possible transitions, and therefore constitutes a smoother estimator.

As a result, we argue that filtering represents the best alternative for continuous gesture recognition and analysis in the context of continuous interaction. We now outline the formal description of the forward algorithm.

FORWARD ALGORITHM Filtering is achieved through the **forward algorithm** (Rabiner, 1989). We define the *forward variable*[6] $\alpha_t(j) = p(z_t = j|\boldsymbol{x}_{1:t})$, that can be computed recursively with a prediction-update cycle:

$$\alpha_1(j) = \frac{1}{Z_1}\pi_j b_j(\boldsymbol{x}_t) \tag{4.7a}$$

$$\alpha_t(j) = \frac{1}{Z_t}\underbrace{\left[\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}\right]}_{\text{prediction}}\underbrace{b_j(\boldsymbol{x}_t)}_{\text{update}} \tag{4.7b}$$

where Z_t is a normalization constant defined by

$$Z_t \triangleq p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) = \sum_{j=1}^{N}\left[\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}\right]b_j(\boldsymbol{x}_t) \tag{4.8}$$

This quantity can be used to determine the likelihood of the observed data given the model's parameters, which is expressed in log form as

$$\log p(\boldsymbol{x}_{1:t}) = \log\left[p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1})p(\boldsymbol{x}_{1:t-1})\right] = \sum_{\tau=1}^{t}\log Z_\tau \tag{4.9}$$

This formula is of major important for classification and continuous recognition as discussed in Section 4.4.3.

6 Note that we define here the forward variable as the filtered state marginal. This is what Rabiner (1989) calls the *scaled* forward variable. This scaling is useful to avoid numerical errors.

4.3.3 Learning

BAUM-WELCH ALGORITHM HMMs can be trained using an Expectation-Maximization (EM) algorithm called the Baum-Welch algorithm. We developed an implementation following the standard algorithm for multiple training sequences and continuous observations with Gaussian mixtures. Therefore, we do not detail the formulation and derivation of the training algorithm, and we refer the reader to (Murphy, 2012, Chapter 15) or Rabiner (1989); Rahimi (2000). Therefore, we do not detail the formulation and derivation of the training algorithm, and we refer the reader to (Murphy, 2012, Chapter 15) or Rabiner (1989); Rahimi (2000).

The Baum-Welch algorithm is an iterative estimation procedure that alternates two steps of *estimation* and *maximization*. In the *estimation* step, we use the current model parameters to compute the smoothed and edged marginals that estimate for each data point the contribution of the states and of the transitions. The *maximization* step re-estimates the parameters based on these intermediate quantities. The equations of the Baum-Welch algorithm ensure that the log-likelihood of the data increases at each iteration, which guarantees the convergence.

CONVERGENCE CRITERION As for Gaussian Mixture Models, we implemented two criteria to define the convergence of the training algorithm. Users can either choose a fixed number of iterations of the Baum-Welch algorithm, or a threshold for the relative percent-change of the log-likelihood. While the former criterion is less relevant in terms of information, it can be advantageous in an interactive machine learning workflow to ensure a constant training time.

INITIALIZATION Initial parameters must be chosen carefully when training with the EM algorithm, in order to avoid convergence to local a maximum. This is all the more important in movement interaction design that interactive applications are usually built from a small set of examples.

Among the common approaches to initial parameter estimation, one can use a mixture model, e.g. a GMM, or a standard K-means algorithm. However, the K-means algorithm with random initialization could also be suboptimal because leading to a local maximum; and the GMM approach does not guarantee the temporal consistency of the clusters with respect to the left-right Markov chain. Although elaborate approaches such as the segmental K-Means algorithm (Juang and Rabiner, 1990) have been proposed, we implemented a more straightforward estimator that is relevant for small training sets.

We propose two approaches according to the observation model:

FULLY OBSERVED APPROXIMATION is used when the observation model is a single Gaussian. Each training example is regularly segmented to estimate the initial values of the states' mean and covariance. In this case, we assume that both states and observations are observed and associated. This approach ensures that the states are initially regularly distributed in time.

GMM-EM is used when using a mixture of Gaussians as observation model. It combines the EM algorithm of GMMs with the previous method. Each segment is therefore associated with a state whose parameters are initially estimated using the EM algorithm for GMMs.

ILLUSTRATIVE EXAMPLE The iterative estimation of the Baum-Welch algorithm is illustrated in Figure 4.11. We train a HMM with 10 states on a two-dimensional eight-shaped gesture drawn with a mouse. The figure compares the results of the EM algorithm for various convergence criteria. On the left, the figure depicts the model after initialization using the Fully Observed Approximation. The states regularly "sample" the example movement. The middle and right plots represent the trained models, respectively after 8 and 83 iterations of the EM algorithm. It is clear that the training converge towards a better approximation of the movement in which the states shift towards the linear portions of the motion parameters.

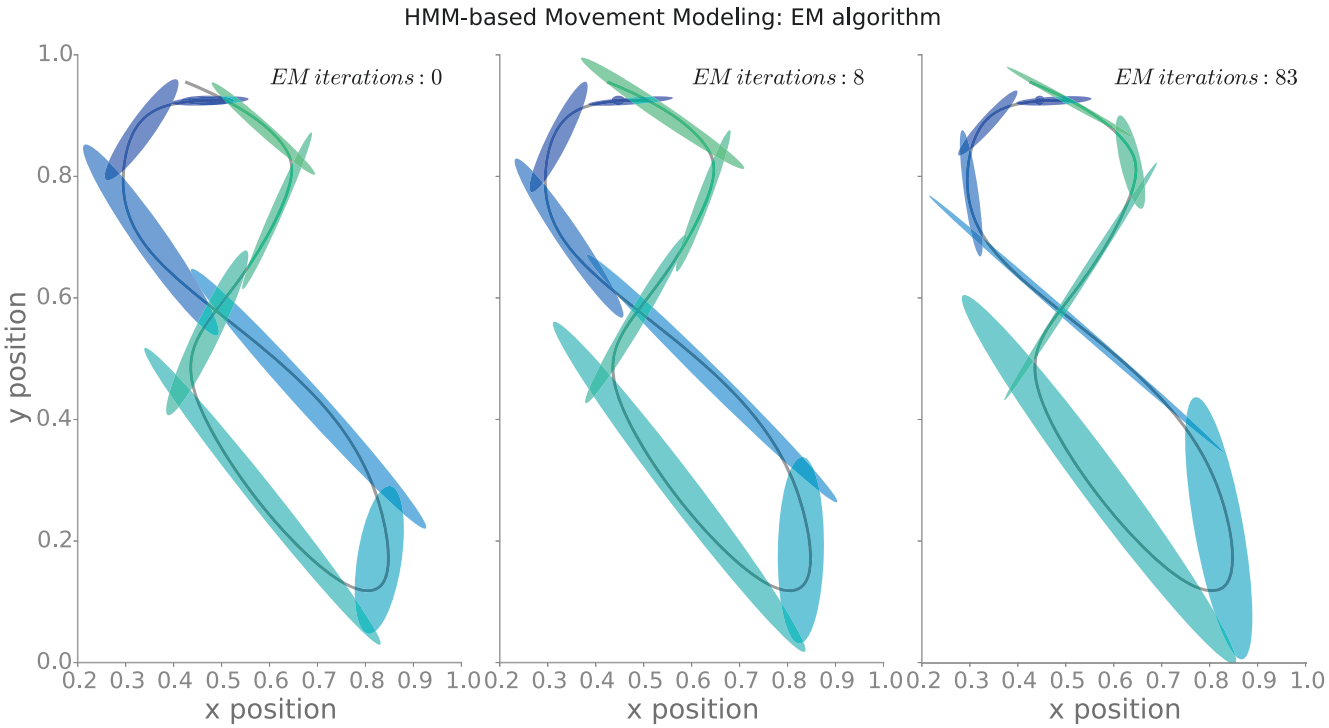
4.3.4 Number of States and Model Selection

The complexity of the model can be adjusted using the number of hidden states. Roughly speaking, it defines how accurately the model is "sampling" or segmenting the training examples.

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.

At the opposite, choosing a large number of states — relatively to the average duration of the training examples, — increases the accuracy of the temporal structure. Nevertheless, can result in overfitting, as the states begin to take into account random variability. Note that in the case of expert and reproducible gestures, overfitting can be an advantage as it provides a very accurate temporal modeling.

MODEL SELECTION Selecting the appropriate number of hidden states can be difficult, and it highly depends on the application. As stressed above, depending on the degree of reproducibility of the users' gestures, we can choose either to embed information in few states or overfit to reach high temporal accuracy. While several methods in the HMM literature address the problem of automatic model selection (see in particular cross-validation and Bayesian Information Criterion), we observed



(a) Representation of the Gaussian states

(b) Structure of the hidden Markov chain

Figure 4.11: Illustration of the EM algorithm for HMM-based movement modeling. A model is trained using a single 2D movement example (solid gray line). The hidden states are represented by their 95% Confidence Interval Ellipse (from blue to green in the left-right order of the hidden chain). The parameters of the EM algorithm are initialized using a Fully Observed Approximation, meaning that each state is initially determined by a regular segmentation of the training example. This estimate is compared with the parameters estimated by the EM algorithm after 8 and 83 iterations. This latter value is defined by a stop criterion on the likelihood, meaning that we consider that the model has converged when the relative change of the log-likelihood becomes inferior to $1e^{-5}$.

that in most cases the most efficient method is the use of direct evaluation with sound feedback.

EXAMPLE Figure 4.12 illustrates the influence of the number of states on the HMM's representation of a gesture. The figure presents three HMMs trained with 5, 10 and 30 hidden states, respectively. Each model was trained on the same single example of 2D movement. Plotting the confidence interval of the distribution of each state illustrates how a small number of states (5 hidden state, left) can result in underfitting, that im-

plies a low accuracy in the temporal modeling. On the contrary, a large number of states (30, right) will ensure a accurate temporal modeling of the gesture's dynamics but is prone to overfitting and will be less tolerant to variations during recognition.

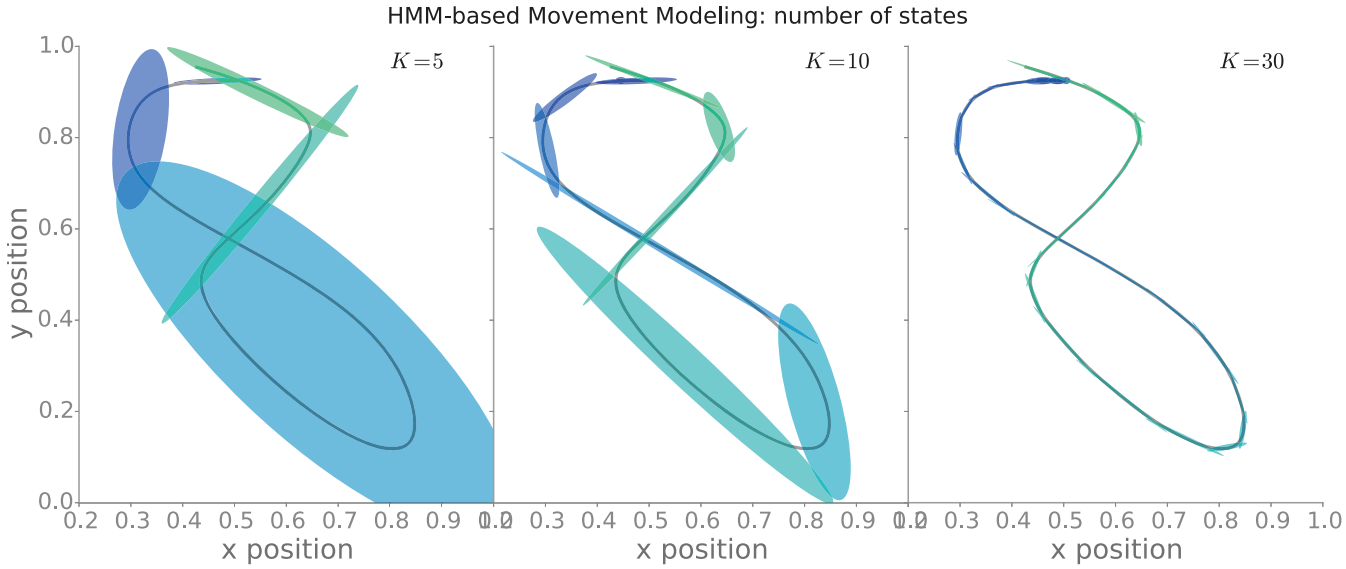


Figure 4.12: Influence of the number of hidden states in HMM-based movement modeling. A model is trained using a single 2D movement example (solid gray line). The hidden states are represented by their 95% Confidence Interval Ellipse (from blue to green in the left-right order of the hidden chain). A low number of states results in underfitting and does't allow for accurate temporal modeling of the movement. A high number of states overfits the example, providing an accurate description of the time profile of the gesture, at the detriment of the generalization of the task.

4.3.5 User-Defined Regularization

Similarly to Gaussian Mixture Models, we introduce a regularization term for the covariances in the update equations of the EM algorithm. After each new estimation of the covariance matrices, the variance of each dimension is incremented by a prior $\lfloor\sigma\rfloor$. The prior is defined similarly to Section 4.1.4, combining a relative prior with an absolute prior. The prior is proportional to the variance of the training set on each dimension, is and thresholded to an absolute minimal value $\lfloor\sigma\rfloor^{(abs)}$, yielding the estimate

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} + \max\left(\lfloor\sigma\rfloor^{(rel)} * Var(\mathscr{X}), \lfloor\sigma\rfloor^{(abs)} * \mathbf{1}_D\right) \cdot \boldsymbol{I}_D \tag{4.10}$$

While regularization is commonly used in HMM implementations to avoid numerical errors and overfitting, it is all the more important in the context of interactive machine learning where few training examples are available. In our implementation, users have access to this parameter that defines the minimal tolerance of the model to new data points.

EXAMPLE We illustrate the influence of the regularization on HMM training in Figure 4.13. We use the same two-dimensional gesture as in the previous examples, and we train two 10-state HMMs with different values for $\lfloor\sigma\rfloor$.

We observe that increasing the regularization artificially increases the zone of influence of the Gaussian component of each state. It is likely that the model trained with a large regularization will be more tolerant to a noisy reproduction of the gesture. However, a too large regularization might reduce the accuracy of a classifier, as it decreases the discriminative power of the model.

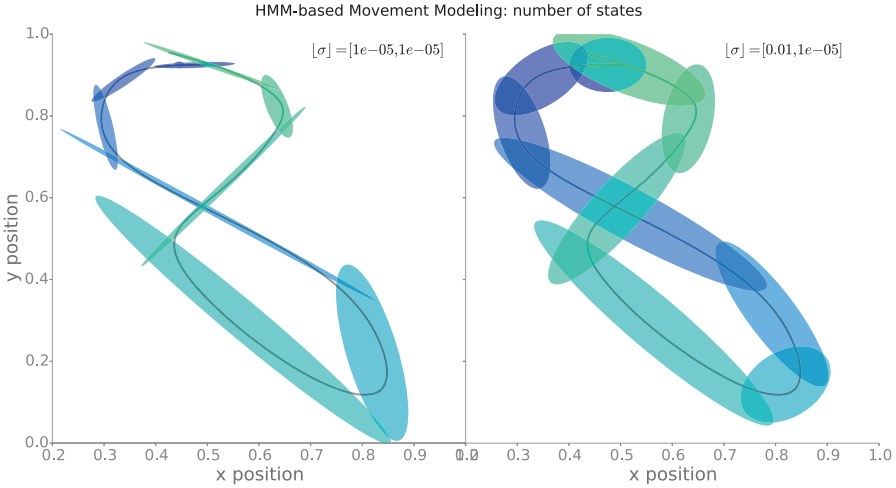


Figure 4.13: Influence of the regularization in HMM-based movement modeling. A model is trained using a single 2D movement example (solid gray line). The hidden states are represented by their 95% Confidence Interval Ellipse (from blue to green in the left-right order of the hidden chain). The figure represents the models learned with different values of the relative part of the regularization. Increasing the prior globally increases the variance over all states, ensuring a better generalization of the movement.

4.4 Temporal Recognition and Mapping with HMMs

We introduced the formalism and implementation of Hidden Markov Models (HMMs) for motion modeling. We now discuss the possibilities offered by the model for mapping design. As for Gaussian Mixture Models (GMMs), we aim to present a simple case of the Mapping-by-Demonstration (MbD) framework that uses continuous gesture recognition and alignment for interacting with recorded sounds. We propose a variation of the *temporal mapping* paradigm introduced by Bevilacqua et al. (2011) for general HMMs.

We argue that while HMMs are often only used for recognition, the model offers a wider range of possibilities through the investigation of its internal parameters. In this section, we aim to formalize a set of design patterns for creating interactions based on continuous recognition parameters.

We start this section by discussing temporal mapping with generic HMMs, and we subsequently discuss the influence of regularization and model complexity. We extend to HMMs the discussion on continuous gesture recognition we presented for GMMs in Section 4.2.

4.4.1 Temporal Mapping

We presented in Section 2.2.3 the *temporal mapping* paradigm formalized by Bevilacqua et al. (2011). Temporal mapping consists in a real-time continuous alignment of a recorded sound on a gesture performance (see Figure 2.1). The process starts with the recording of a movement performance synchronized with an audio recording. In an MbD context, the gesture can be performed while listening to the sound example. In performance, we compute a continuous alignment of a new gesture over the reference, and we accordingly re-align the audio recording.

The method proposed by Bevilacqua et al. (2011) uses Gesture Follower to estimate the time progression of a gesture within a template recording. In Gesture Follower, a HMM is built by associating a state to each frame of the reference recording. At runtime, they use the index of the likeliest state to compute the temporal position within the template.

GESTURE PROGRESSION We propose to generalize this method to an HMM with an arbitrary number of hidden states, and possibly trained on several recordings of the same gesture. At each new observation, we estimate the normalized time progression within the model as the expected value of the state posterior probabilities:

$$\begin{aligned} \bar{\tau}(\boldsymbol{x}_t) &\triangleq \frac{1}{N-1}\mathbb{E}[\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}] \\ &= \frac{1}{N-1}\sum_{i=1}^{N}(i-1)\alpha_t(i) \end{aligned} \tag{4.11}$$

where the states z_t are indexed from left to right, and their probability is defined by the forward variable α. The process is illustrated in Figure 4.14.

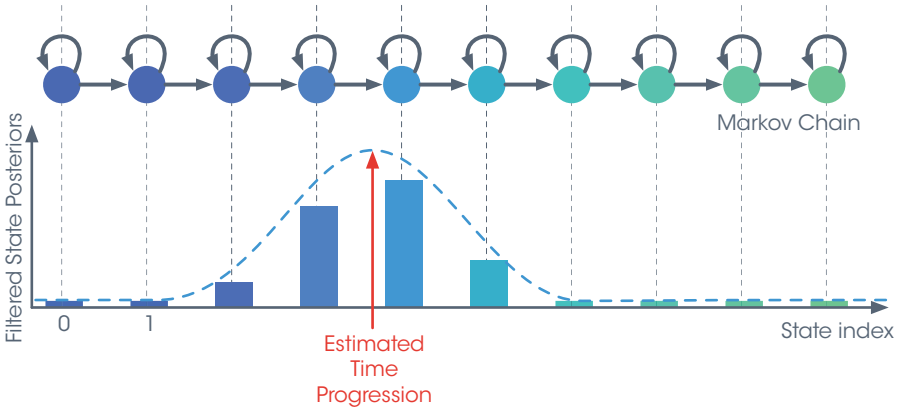


Figure 4.14: Estimation of the time progression as the expectation of the filtered state posteriors.

In Gesture Follower, each state is associated to a frame of the reference performance, which brings a correspondence between the index of the states and the absolute time of the gesture performance. In our case, we use a reduced number of hidden states to embed the temporal structure of the gesture. The progression represents a relative progress of the observed gesture within the model, that depends on the information contained in each state. Notably, it is important to note that all states might not encode the same duration. Therefore, the progression relates to the information content of the sequence of states — in particular the variation in dynamics of the motion parameters, — rather than to the absolute time scale of the performance.

4.4.2 User-defined Regularization and Complexity

NUMBER OF HIDDEN STATES The number of states determines the accuracy of the temporal structure of the modeled movement. For temporal mapping, choosing the appropriate number of hidden states is therefore crucial to reach the desired accuracy and smoothness in the alignment.

As an example, we consider the eight-shaped gesture presented in the example of Figure 4.12. We propose to investigate how the number of hidden states impacts on the structure of the model for continuous alignment. To illustrate this, we train a model with a single gesture example. In a second step, we use the same recording in 'follow' mode, to illustrate how the model estimates the gesture progression.

Figure 4.15 depicts the state probability distribution along the gesture, as well as the estimated progression. Using 6 states gives a very rough representation of the gesture's temporal structure. It highlights a very unequal repartition of the states in time, that gives a segmentation of the gesture. As there is very few overlap between the Gaussian component of each state, the alignment is a step function. Increasing the number of states improves the resolution of the alignment, and therefore provides a more accurate estimate for sound control.

REGULARIZATION We introduced in Section 4.3.5 a regularization strategy that artificially increases the variances of the states through a prior on the covariance matrices. We now give insights into how users can exploit regularization as a critical parameter for continuous gesture alignment. We argue that it allows us to 1) artificially increase the generalization of a model trained on a single instance and 2) smooths the estimation of the temporal alignment.

Figure 4.16 illustrates the influence of regularization on continuous gesture following. As in the previous examples, we use the same eight-shape gesture for both learning and following, in order to illustrate the properties of the progression estimate. The figure illustrates that using a large regularization increases the overlap between the states, which results in a smoother estimation of the progression.

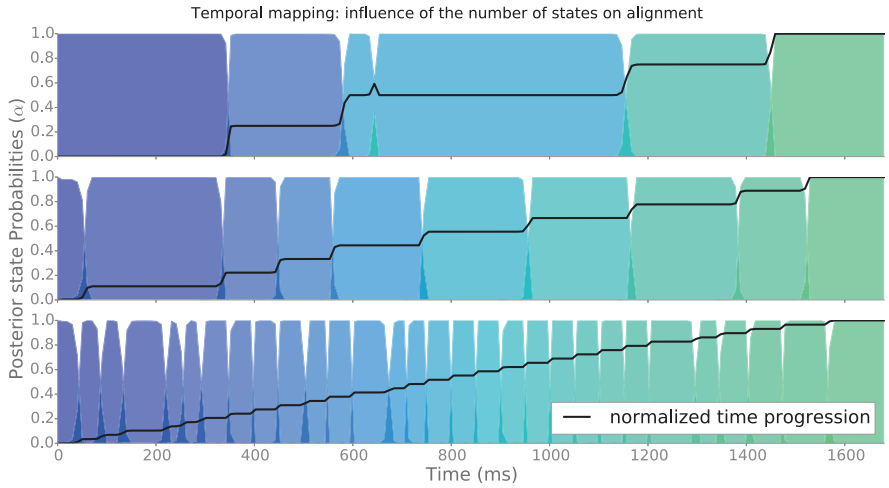


Figure 4.15: Influence of the number of hidden states on real-time alignment. We use the same eight-shaped gesture as in the previous examples. A single instance of a gesture is used to train a HMM, and we perform the alignment using the same example. In the figure are plotted the posterior probabilities of each state (shaded areas, from blue to green according to the order of the state in the left-right chain). The normalized time progression is depicted by the solid black line. The models were trained 6, 10 and 20 states, respectively, and regularization $\lfloor\sigma\rfloor = [1e-5, 1e-5]$

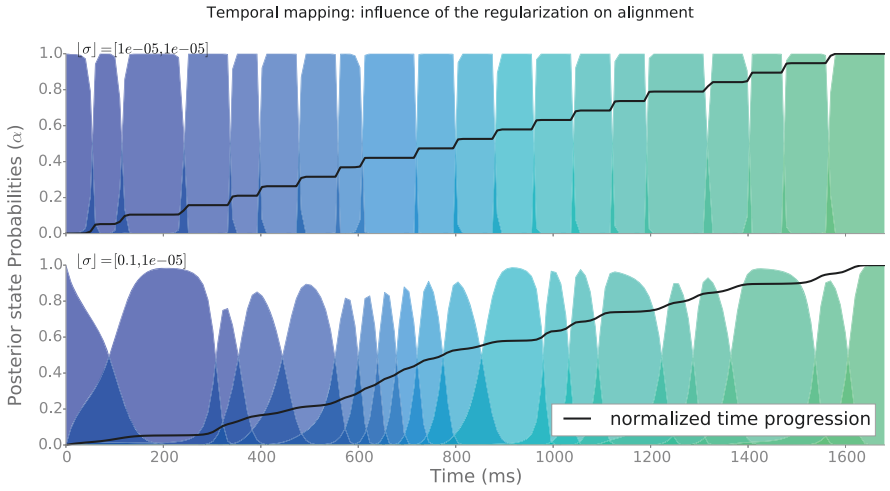


Figure 4.16: Influence of the regularization on real-time alignment. We use the same eight-shaped gesture as in the previous examples. A single instance of a gesture is used to train a HMM with 20 states, and we perform the alignment using the same example. In the figure are plotted the posterior probabilities of each state (shaded areas, from blue to green according to the order of the state in the left-right chain). The normalized progression is depicted by the solid black line. We compare two values of the relative regularization: $1e-5$ and $1e-1$, respectively.

4.4.3 Classification and Continuous Recognition

Temporal mapping can be combined with gesture recognition to associate various gestures to different audio recordings. In this case, we extend the

paradigm by using continuous gesture recognition to select or mix the appropriate sounds. Note that gesture recognition can also be used independently from alignment where the recognized gesture serves to trigger the audio samples.

We discussed classification and continuous recognition for Gaussian Mixture Models in Section 4.2.2. The same distinction applies to Hidden Markov Models, and the recognition process is expressed in the same way using class-conditional likelihoods. For online recognition, we use a forward algorithm that estimates the likelihood of each class causally using Equation 4.9. The posterior class likelihoods can be derived using Bayes rule, yielding a normalized estimate for each class:

$$p(c \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid c)}{\sum_{c'=1}^{C} p(\boldsymbol{x}|c')} \tag{4.12}$$

This formulation allows us to develop several strategies, such as triggering/selection based on classification — selecting the class maximizing the posterior likelihoods; — or using the continuous variations of the log-likelihoods or posterior likelihoods as a continuous sound control.

It is important to highlight that HMMs introduce a sequence model that impacts the estimation of the likelihoods. While for GMMs the likelihood is computed on a frame-by-frame basis, in HMMs it depends on the whole history of the movement stream, as schematized in Figure 4.17. Therefore, it integrates both short-term and long-term dependencies in the estimation of the likelihoods, which makes HMMs more robust for dynamic gesture recognition.

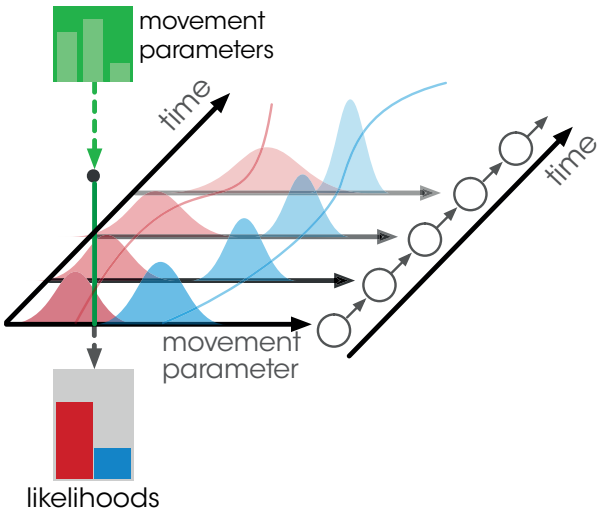


Figure 4.17: Schematic representation of recognition with Hidden Markov Models.

4.4.4 Discussion

We presented a variation of the temporal mapping paradigm using generic Hidden Markov Models (HMMs) where audio is dynamically aligned to the

performance of the gesture. This led us to formalize a set of interaction design patterns based on HMMs, that we summarize in Figure 4.18.

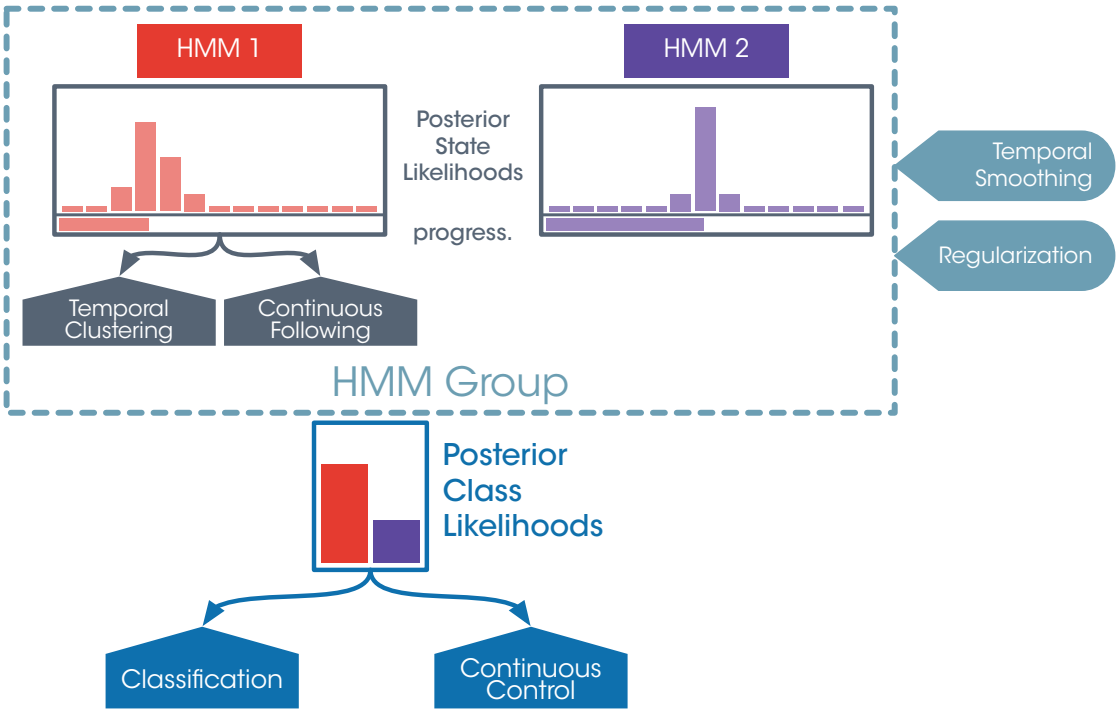


Figure 4.18: Summary of the Design Strategies based on Hidden Markov Models.

Investigating the recognition parameters of the model allows us to develop elaborate strategies for designing sonic interactions. In particular, the class posteriors can be used for continuous control of sound parameters, to go beyond simple gesture classification. We showed that the user-defined parameters of number of states and regularization are essential for adjusting the behavior of the model for continuous interaction.

4.5 Segment-level Modeling with Hierarchical Hidden Markov Models

We presented a movement modeling approach based on Hidden Markov Models (HMMs) where each gesture is encoded in a single left-right model. Associating a model to each gesture class allows us to perform real-time recognition by comparing the class-conditional likelihoods. At runtime, the models are therefore running independently of each other.

This independence between several classes can be limiting for continuous gesture recognition. As a matter of fact, real-world problems often involve performing recognition from a continuous stream of motion where gestures are continuously sequenced. Such situations involve that gestures are not independent of each other, as their ordering might be determined by contextual constraints. Such constraints often result in co-articulation, as subsequent gestures overlap and influence each other. Many psychological studies support a temporal representation of gestures as a sequence of phases:

> A *Gesture Phrase* may be distinguished, thus, as a *nucleus* of movement having some definite form and enhanced dynamic qualities, which is preceded by a preparatory movement and succeeded by a movement that either moves the limb back to its rest position or repositions it for the beginning of a new Gesture Phrase. (Kendon (1986))

Such representations have already been applied in computational models (Pavlovic et al., 1997), and are all the more important in music, where multilevel structures are ubiquitous. In terms of modeling, this require integrating high-level structures in the representation of gestures in order to account for these long-term dependencies.

LIMITATIONS OF HMMS In HMMs, observations are produced at the frame level. The conditional independence assumption between successive observations is therefore limiting the representation of high-level features and long-term dependencies. For example, there is no mechanism for supporting the transitions between successive segments, for example to 're-initialize' the recognition to an initial state when we reach the end of a gesture. Moreover, we presented a modeling technique that considers gestures as continuous, unbreakable units, that therefore fail to address more modular representations of gestures as proposed by Kendon (1986).

SEGMENTAL AND HIERARCHICAL HMMS These issues have been addressed through various extensions of HMMs. Hidden Semi-Markov Model (HSMM) introduce an explicit distribution over the durations of the hidden states (Murphy, 2002a; Yu, 2010). In the Segmental Hidden Markov Model (SHMM), each hidden state emits a sub-sequence of observations rather than a single one, given a geometric shape and a duration distribution. The model was applied to speech recognition (Ostendorf et al., 1996), handwritten shape recognition (Artières et al., 2007) and, at Ircam, time profile recognition of pitch and loudness (Bloit et al., 2010) and segmentation of musical gestures (Caramiaux et al., 2011). As argued in previous work (Françoise, 2011), the SHMM is limited by its representation of gesture segments. Indeed, in the model each segment is represented by a template shape that can be stretched uniformly according to the duration distribution. However, timing often evolves in complex ways in gesture performance, often with local variations.

Alternatively, we propose to use the Hierarchical Hidden Markov Model (HHMM), that provides a more flexible framework for movement modeling. The HHMM (Fine et al., 1998) extends the standard HMM by integrating a multi-level representation of gestures. The model is built upon a hierarchy of hidden states, where each state generates a sub-model, forming a tree structure. This model has the temporal flexibility of HMMs while providing an arbitrarily deep hierarchical structure governing higher level transitions.

In this section, we outline the representation, inference and learning issues for the HHMM. For an extensive study of the model with applications to gesture recognition and segmentation, see Françoise (2011).

4.5.1 Representation

OVERVIEW We now define a gesture (or movement) as a sequence of "motion segments", where each segment is a continuous time profile of motion parameters. In the method presented in Section 4.3, each segment is represented by a single left-right HMM, and the models are running independently. In the HHMM, if the segments are modeled in the same manner using left-right HMMs, they are now embedded in a higher-level probabilistic transition structure. For this purpose, we add on top of the existing models a new layer of hidden states that defines the transition structure between the models.

As presented by Fine et al. (1998) and Murphy and Paskin (2001), the model can easily be extended to an arbitrary number of levels. In this case, the model has a tree structure that defines a hierarchy of states from a single root to the leaf states which emit observations. In this work, we are interested in modeling gestures at the segment level. Therefore, we only consider the case of Hierarchical HMMs with two levels.

FORMAL DESCRIPTION The topological representation of a HHMM with three motion segments is depicted in Figure 4.19. We define two types of hidden states:

SIGNAL STATES compose the lower level that emits observations similarly to the hidden states of a HMM: their observation models are directly associated with motion data and use Gaussian distributions. This *signal* level encodes the fine temporal structure of the segment.

SEGMENT STATES are associated with labeled motion segments. Instead of directly emitting observation, these *internal* states generate the sub-models of the *signal* level. Segments states can be seen as producing sequences of observations through the activation of their sub-model.

Consider a sequence of observations $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$, and the associated sequence of signal states z_t and segment states s_t. Formally, a HHMM with M segments is defined by the set of parameters $\lambda = \left\{\boldsymbol{H}, \boldsymbol{G}, \{N^{(m)}, \boldsymbol{\pi}^{(m)}, \boldsymbol{A}^{(m)}, \boldsymbol{B}^{(m)}\}_{m=1}^{M}\right\}$ composed of the parameters of each signal-level model m (as defined in Section 4.3.1), augmented with the prior vector $H = \{h_m\}$ and the state transition matrix $G = \{g_{ml}\}$ of the segment level where:

$h_m \triangleq p(s_1 = i)$ is the prior probability of the mth segment state.
$h_m \geq 0$ and $\sum_{m=1}^{M} h_m = 1$

$g_{ml} \triangleq p(s_t = l | s_{t-1} = m)$ is the probability of transiting from state m to state l.
$g_{ml} \geq 0$ and $\sum_{m=1}^{M} g_{ml} = 1$

Note that the parameters of each sub-model need to be expressed conditionally to their *parent segment* state as

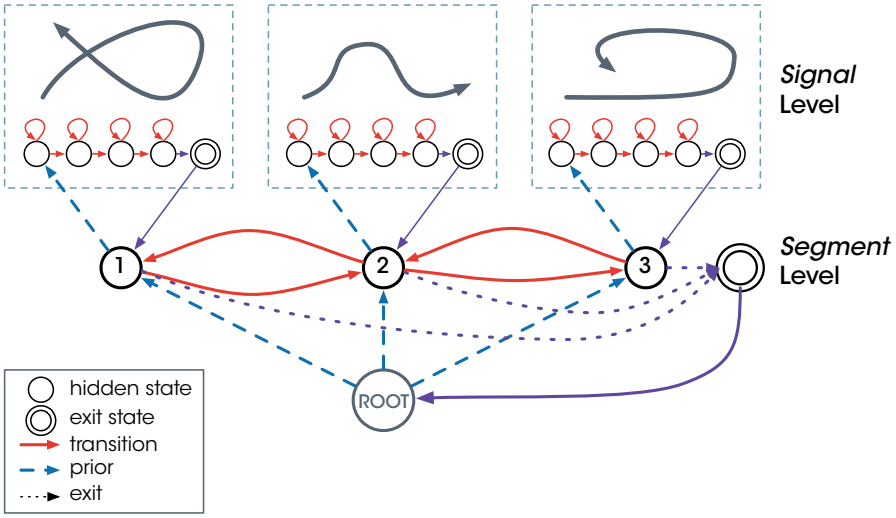


Figure 4.19: Graphical Model of a 2-level HHMM. The model represents 3 motion segments. As for the standard HMM representation, each segment is modeled sing a left-right HMM. These *signal*-level models are embedded in a higher-level structure (*segment* level) that has its own transition structure. An *exit* mechanism us allows us to define the probability to terminate from each state,to go back to parent level in order to make a transition. This mechanism is represented by double-edged circles.

$$\begin{aligned} \pi_i^{(m)} &\triangleq p(z_1 = i | s_1 = m) \\ a_{ij}^{(m)} &\triangleq p(z_t = j | z_{t-1} = i, s_t = m) \\ b_j^{(m)}(\boldsymbol{x}_t) &\triangleq p(\boldsymbol{x}_t | z_t = j, s_t = m) \end{aligned}$$

To guarantee the consistency of the model, we need to add a new type of state at each level of the hierarchy: an *exit* state that allows to go back to the parent level in order to make a transition.[7] We define the vectors $\boldsymbol{a}_{exit}^{(m)}$ and $\boldsymbol{g}_{exit}$ which encode the probabilities for each state of a given level to reach the exit state:

$a_{exit,i}^{(m)}$ is the probability to exit the signal model m from its state i

$g_{exit,m}$ is the probability to exit the segment state m to go back to the root.

The normalization of the transition matrices must now take into account these *exit* probabilities, as the probability of transiting from a state must sum to one:

$$\begin{aligned} \textstyle\sum_{i=1}^{N^{(m)}} a_{ij} + a_{exit,i}^{(m)} &= 1 \\ \textstyle\sum_{l=1}^{M} g_{lm} + g_{exit,l} &= 1 \end{aligned}$$

The joint distribution of the model takes a complex form, because of the cross-dependency between *signal* and *segment* states: transitions at the *signal* level are conditioned on transitions at the *segment* level, while transition of the *segment* level depends on the possibility to exit the sub-

7 Note that the *exit* states are 'virtual' states, as they do not emit observation in any manner. Their role is to favor the transitions from a set of exit points with a segment.

model at the *signal* level. Additional representation details can be found in Françoise (2011).

4.5.2 Topology

Our goal in modeling movement trajectories is to articulate a continuous representation of motion primitives with higher-level sequencing issues. As the *signal* level encodes the temporal structure of continuous motion trajectories, we defined a transition structure respecting temporal constraints, through a left-right topology. The *signal*-level parameters take the form:

$$\begin{aligned} \pi_i^{(m)} &= \delta(i,1) \\ a_{ij}^{(m)} &= a_{ii}\delta(i,j) + (1-a_{ii})\delta(i,j-1) \\ a_{exit,i}^{(m)} &= a_{exit}\delta(i,N^{(m)}) \end{aligned}$$

No assumption is made for the topology of the *segment* level, that is assumed ergodic in a generic case, allowing equal probability to all possible transitions. The design and learning of the high-level transition matrix is discussed in Section 4.5.4, and an example is given in Section 4.6.2.

4.5.3 Inference

We already discussed the types of inference for HMMs in Section 4.3.2. A similar argument can be made for the Hierarchical HMM. For the purpose of real-time continuous interaction, filtering is the most efficient method to estimate state probability densities in a causal way. For the HHMM, filtering means estimating both the *signal* and *segment* states from the observed sequence of movement features $p(z_t, s_t|\boldsymbol{x}_{1:t})$.[8]

In a seminal article, Fine et al. (1998) proposed a set of inference algorithms derived from the *input-output* algorithm for Stochastic Context-Free Grammars. However, the algorithm's cubic complexity in the length of the observation sequence makes it intractable for both offline and online inference. A more efficient solution proposed by Murphy and Paskin (2001) consists in representing the HHMM as a Dynamic Bayesian Network (DBN) (Murphy, 2002b).

DYNAMIC BAYESIAN NETWORK REPRESENTATION

Dynamic Bayesian Network are a special case of Bayesian Networks for modeling sequential data. In a DBN, the internal state of a system at a given instant is represented by a set of hidden variables, complemented with input and observed variables within a *slice*. Each time slice is conditioned on the slice at the previous time step. The simplest DBN is the Hidden Markov Model (HMM) that has a single hidden variable per time slice (see Figure 4.9).

8 In this work we only consider filtering. However, we previously studied and compared filtering, fixed-lag smoothing and MAP estimation for continuous gesture segmentation and recognition (Françoise, 2011).

Figure 4.20 shows the representation of a 2-level Hierarchical HMM as a DBN. In the graph, the conditional probability distributions are represented by arrows. The transition structure at both the *signal* and the *segment* level are represented by horizontal arrows, while vertical arrows express the conditioning of signal states z_t on a parent *segment* state. Additionally, *exit* states are represented by the binary indicator variables F_t and U_t where

$$\begin{aligned} p(F_t = 1 \mid s_T = m, z_t = i) &= a^{(m)}_{exit,i} \\ p(U_t = 1 \mid S_t = m, F_t = f) &= g_{exit,m}\delta(f,1) \end{aligned}$$

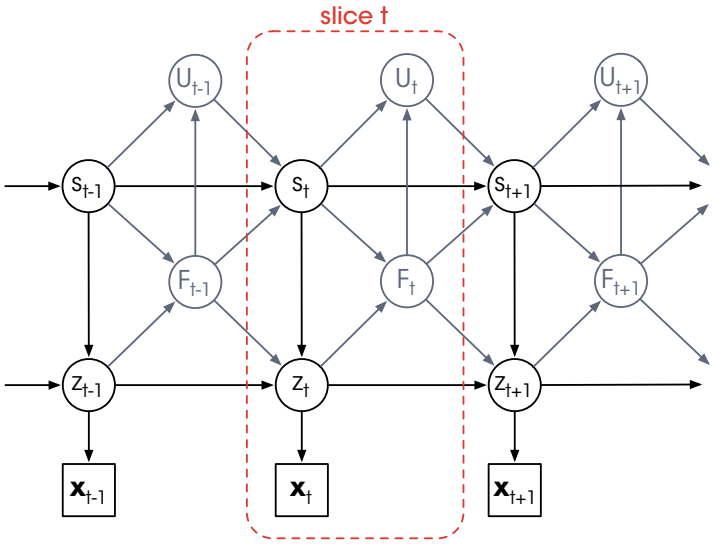


Figure 4.20: Dynamic Bayesian Network representation of a 2-level HHMM. Hidden variables are represented by circle nodes, square nodes represent observed variable, arrows indicate dependencies. s_t and z_t represent respectively *segment* and *signal* states, whereas the variable F_t and U_t are binary indicators representing the possibility of finishing at their respective level to make a higher-level transition.

FORWARD ALGORITHM As for HMMs, filtering is achieved through the forward algorithm that estimates the joint probability of all hidden variables given the causal sequence of observations. For the 2-level HHMM, we define the forward variable $\alpha_t(j,m) = p\left(z_t = j, s_t = m \mid \boldsymbol{x}_{1:t}\right)$ which can be computed recursively through a prediction-update cycle:

$$\alpha_t(j,m) = \frac{1}{Z_t} b^{(m)}_j(\boldsymbol{x}_t)\left[T^{ext}_t(j,m) + T^{int}_t(j,m)\right] \tag{4.13}$$

where T^{int}_t and T^{ext}_t represent respectively the transitions from a state within the same segment, or from a state of another segment:

$$T^{int}_t(j,m) = \sum_{i=1}^{n^{(m)}}\left[a^{(m)}_{ij} \cdot \alpha_{t-1}(i,m)\right] \tag{4.14a}$$

$$T^{ext}_t(j,m) = \pi^{(m)}_j \sum_{l=1}^{M}\sum_{i=1}^{N^{(l)}} a^{(l)}_{exit,i}\left[g_{lm} + g_{exit,l}h_m\right]\alpha_{t-1}(i,l) \tag{4.14b}$$

and where Z_t is a normalization constant:

$$Z_t \triangleq p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) = \sum_{m=1}^{M} \sum_{j=1}^{N^{(m)}} b_j^{(m)}(\boldsymbol{x}_t) \left[T_t^{int}(j,m) + T_t^{ext}(j,m) \right] \tag{4.15}$$

4.5.4 Learning

Learning hierarchical models from unlabeled data is challenging. Here, we focus on supervised learning problems where the training data is labeled in classes. Therefore, we can train the Hierarchical HMM in a semi-observed setup, where each segment in the training set is associated to a class. This makes it possible to train each sub-modal independently for all the classes of the training set.

The training is similar to standard HMMs: we use an EM algorithm to estimate the *signal*-level parameters of each segment. Learning the segment level could be done using an EM algorithm. However, this requires an important set of training examples containing long sequences of motion segments.

Alternatively, we chose to let the high-level structure ergodic by default — allowing all segment-level transition with equal probability, — but editable manually. Authoring this high level transition structure allows users to define particular vocabularies governing the transition between gestures, for example answering some compositional constraints; or enables to create new representations of gestures where the transitions between motion segments are constrained, as we propose in Section 4.6.2

4.5.5 Discussion

The Hierarchical Hidden Markov Model (HHMM) brings a higher level representation of gestures through a transition structure governing the sequencing of motion segments. It is always possible to represent a HHMM as a HMM, by flattening its structure to form a fully connected HMM. However, as argued by Murphy and Paskin (2001), by doing this we loose the advantages of the hierarchical structure, that allows an easier setting of the segment transitions. Moreover, the exit probabilities define a transparent mechanism for initiating transitions to new motion segments when the current segment is ending. The Dynamic Bayesian Network (DBN) representation provides efficient inference algorithms while preserving the hierarchical structure.

4.6 Segment-level Mapping with the HHMM

This section presents a central contribution of this chapter: the extension of sound control strategies to a segmental representation. We propose to improve the temporal structure of the sound synthesis by integrating a representation of gestures and sounds as sequences of segments. This formal-

ism allows us to develop sound control strategies that preserve the transients of the sounds in performance.

This section is an adaptation of a previous publication: "A Hierarchical Approach for the Design of Gesture-to-Sound Mappings", presented at the Sound and Music Computing conference in 2012 (Françoise et al., 2012). For clarity, and to avoid interactions with the theoretical aspects of the modeling framework, we chose to adapt the article rather than reporting the full publication.

We start by discussing how the high level transition structure Hierarchical Hidden Markov Model (HHMM) improves continuous gesture recognition. Then, we propose a representation of gestures designed for the control of recorded sounds, as a sequence of four phases: Preparation-Attack-Sustain-Release. Finally, we discuss how such a representation improves the temporal structure of the sound synthesis in a Mapping-by-Demonstration framework.

4.6.1 Improving Online Gesture Segmentation and Recognition

In this section we build upon the joint segmentation and recognition strategy presented for HMMs in Section 4.4.3. We argue that the HHMM offers a more efficient and accurate way to perform recognition in real-time, thanks to its high-level structure. Following equation 4.13, we obtain the joint likelihood of class c by marginalizing the state probabilities over the *signal* states:

$$p(s_t = c, \boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) = \sum_{j=1}^{N^{(c)}} b_j^{(c)}(\boldsymbol{x}_t) \left[T_t^{int}(j,c) + T_t^{ext}(j,c) \right] \tag{4.16}$$

and the posterior class densities can be expressed directly from the forward variable as

$$p(s_t = c \mid \boldsymbol{x}_{1:t}) = \sum_{j=1}^{N^{(c)}} \alpha_t(j,c) \tag{4.17}$$

Several mechanisms of the hierarchical representation may improve the accuracy of the segmentation. First, the posterior state probabilities are scaled globally, which strengthens the discriminative quality of the model. When entering a segment with a high degree of certainty, the likelihood of other segments decreases significantly.

Second, the model integrates an *exit* mechanism that is crucial for continuous online recognition. The exit probabilities in a left-right model are non-zero only on the last state of the segment. This means that when a motion segment is recognized, the probabilities of exiting the segment will increase as the last states of the segment accumulate probabilities. When these probabilities become large enough, they allow a transition at the *segment* level — this is expressed by the exterior transition term T^{ext}, — which re-distributes probabilities on the accessible segments.

We further compare GMMs, HMMs and HHMMs in Chapter 5 for continuous recognition and alignment.

4.6.2 A Four-Phase Representation of Musical Gestures

We present in this section an example of decomposition of gestures as ordered sequences of motion segments. The representation draws from the formalism of Kendon (1986), who proposed that gestures might be composed of a preparation, followed by a nucleus and a relaxation gesture. Inspired from the classical ADSR representation of sound envelopes — standing for *Attack*, *Decay*, *Sustain*, *Release*, — we introduce, as an example, a decomposition of gestures into four typical phases in gestures for sound control, defined as follows:

- **Preparation (P):** anticipation gesture preceding the beginning of the sound.
- **Attack (A):** segment covering the attack transient of the sound.
- **Sustain (S):** segment spanning from the decay to the end of the sound.
- **Release (R):** retraction gesture following the end of the sound.

REPRESENTATION USING THE HIERARCHICAL HMM

Such a representation can be effectively and efficiently implemented using the proposed two-level HHMM. Our implementation allows users to author the high level transition structure, making for example some segments optional (such as the preparation or release) or imposing constraints on segment ordering.

Figure 4.21 illustrates a possible topology for representing gestures as PASR. The *segment* states are S_1 = P, S_2 = A, S_3 = S and S_4 = R, and the parameters of the model are set to allow transitions in the sequential order. The segment-level prior probabilities are equally set to 0.5 on the P and A states, ensuring that the gesture can be entered equally through the preparation or the attack phase. Within the gesture, transitions are defined from left to right to respect the sequential order. Finally, additional probabilities have to be set, which define the possibility of reaching the *exit* state — represented by a double circle on the figure — and go back to the root in order to enter another gesture. These probabilities are equal to 0.5 and 1 for the last two states of the model, restricting the possibility of ending a gesture through the sustain phase or the release phase.

Therefore, two modes are possible when performing sequences of gestures. Each gesture can be performed entirely, from the preparation to the release, or can be sequenced in a shorter form by avoiding the preparation and release segments. Thus, different transitions between gestures are possible.

In Figure 4.22, we show an example of the decomposition of a complex gesture based on two gestures templates. On the top left of Figure 4.22, two different gesture templates are learned. Both are decomposed into the 4 phases P, A, S, and R, which define the topological structure of the two-level Hierarchical HMM, as previously introduced by Figure 4.21.

On the top right part of the figure, an input gesture is decomposed using the two templates. The inference process segments the input gesture and

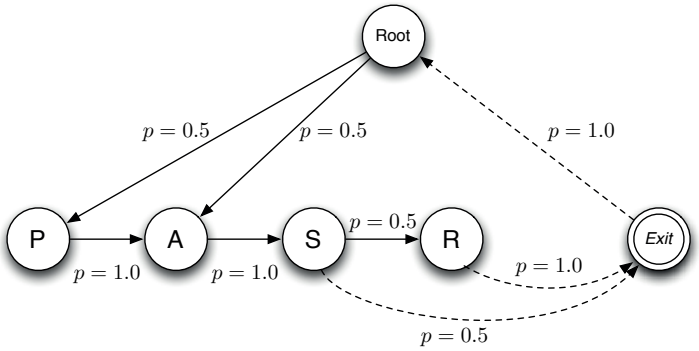


Figure 4.21: Topology of the PASR gesture models for 1 gesture. The prior probabilities ensure that the gesture can only be entered by the *Preparation* or *Attack* phases. Gesture models are left-to-right and reaching the exit state is only possible from the *Sustain* and *Release* phases.

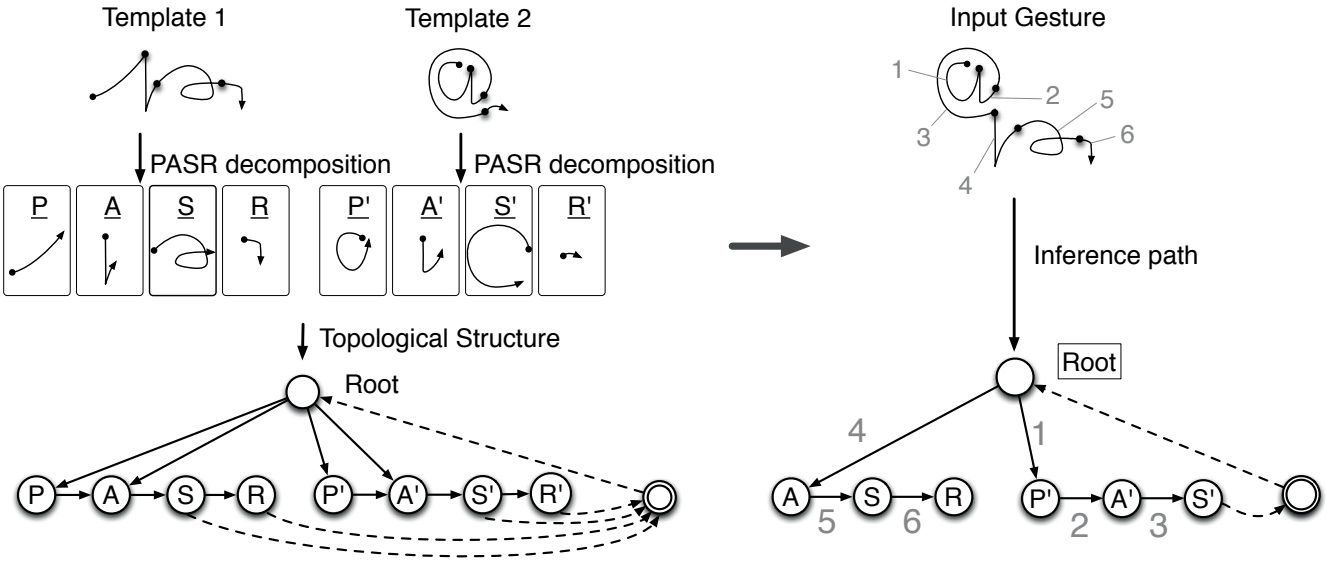


Figure 4.22: A practical example of the PASR decomposition of gestures. Two template gestures can be learned, represented at the top left of the figure. The decomposition of each gesture defines the structure of the model (bottom left). During performance, a continuous gesture can be performed by sequencing several segments of the original templates (top right). This induces a specific path in the topological graph (bottom right).

recognizes the gesture segments. This induces a path in the topological graph, depicted on the bottom right of Figure 4.22. Note that this type of information can be computed in real-time due to the forward inference.

4.6.3 Sound Control Strategies with Segment-level Mapping

The PASR structure allows us to derive elaborate techniques for gestural interpretation of recorded sounds. In particular, we extend the temporal mapping paradigm proposed by Bevilacqua et al. (2011). Our approach

aims to provide a more non-linear way of replaying sounds through a modular representation of gestures in relationship to sounds. The PASR structure also improves the quality of the sound synthesis on transients. This work was previously published at the Sound and Music Computing Conference in 2012 (Françoise et al., 2012).

APPLICATION ARCHITECTURE Although other types of sensors could be used with the system, in this application we focus on inertial sensors. In particular, we use the MO interfaces developed at Ircam, that embed a 3D accelerometer and a 3 axis gyroscope (Rasamimanana et al., 2011).

Figure 4.23 details the general workflow of the application, and a screenshot of the Max patch is reported in Figure 4.24. The patch provides visualization and editing tools for both sounds and gesture signal, coupled with a control panel. The control panel can be used to add or remove buffers, save and load presets, and play the sound (top of Figure 4.24).

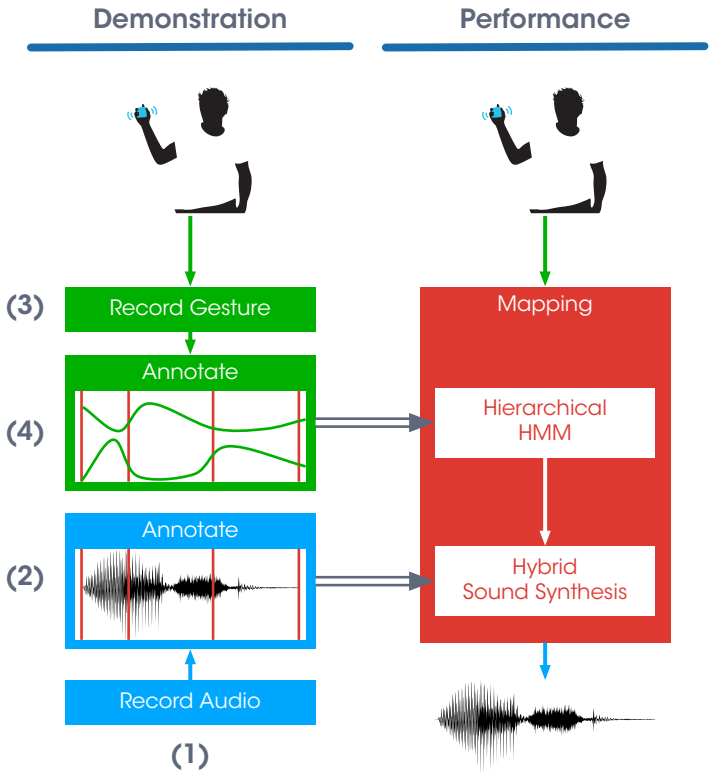


Figure 4.23: Workflow diagram of the application

We describe below first the demonstration mode, necessary to build the hierarchical gesture models from templates recorded by the user. and second, the performance mode, where the gesture segmentation and recognition process drives phase vocoder sound processing.

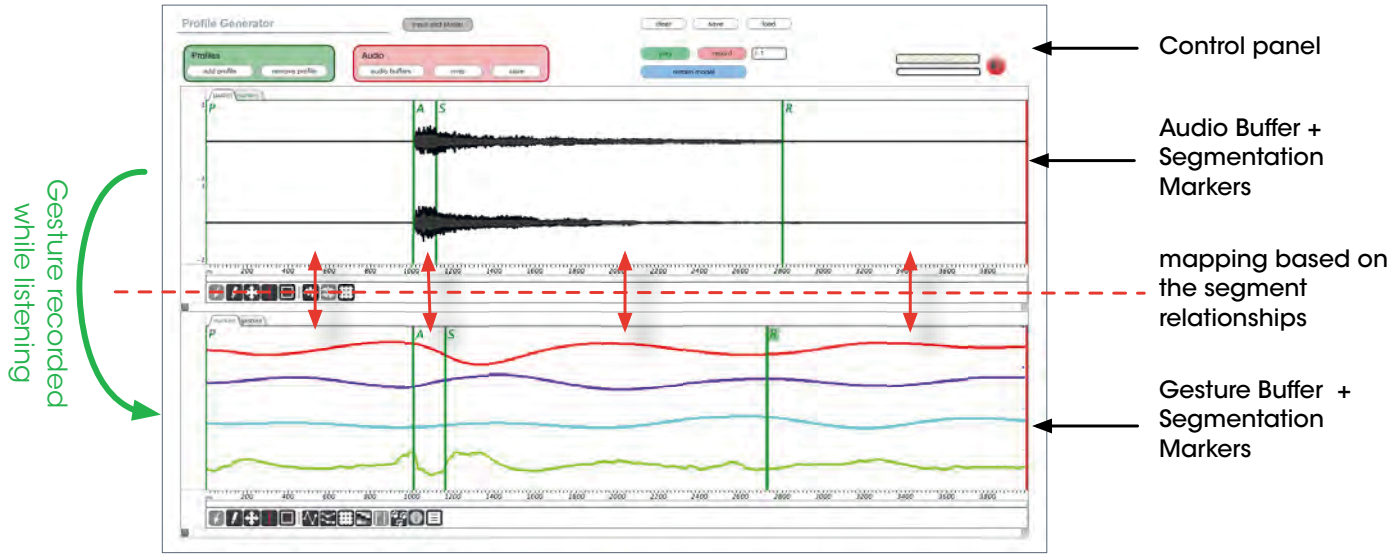


Figure 4.24: Screenshot of the Max Patch of the application.

DEMONSTRATION In the proposed application, the gesture segmentation is computed using the two-level Hierarchical HMM introduced in the previous sections. The model has been implemented as an external object for Max allowing to perform the multilevel gesture segmentation in real-time.

A sound is represented by its waveform at the bottom of Figure 4.24. First, the user must add markers and label the segmentation on the audio buffer, to define the audio segments that will be linked to the gesture segments: Preparation, Attack, Sustain and Release (PASR) (phase (1) in Figure 4.23).

Second, the user must perform a gesture, where the PASR decomposition can be operated. One possible strategy is to perform the gesture while listening to the sound, in order to induce structural similarities with the audio sample. This gesture is recorded in a gesture buffer, as shown at the bottom of Figure 4.24. As with the sound buffer, the gesture data must be annotated with a set of markers defining the P, A, S and R phases of the gesture (phase (3) in Figure 4.23). If the gesture was performed synchronously with the sound, the markers can be transferred from the audio buffer and re-edited to closely fit the timing of the gesture performance. Finally, the segmented gesture can be used to build the hierarchical model (phase (4) in Figure 4.23), and specific messages are used to set the high level parameters (e.g. prior, transition, and exit probabilities) as specified in section 4.6.2, with respect to the PASR decomposition.

Finally, the user can switch to the performance mode and evaluate the quality of the control. At any moment, she can switch back to the demonstration mode to adjust the examples and train the model.

PERFORMANCE In performance mode (Figure 4.23, the stream of motion parameters is segmented and labeled automatically. The recognition and temporal alignment of the motion segments is then used to control sound synthesis.

Precisely, the Max object outputs a set of parameters at each new observation: the likelihood of each gesture segments, the time progression and the estimated speed of the input gesture compared with the templates. Therefore, the object continuously updates the following information: the index of segment currently performed and the temporal position within the segment.

This information is used to control temporal dynamics of recorded sounds, mixing sampling and phase vocoder techniques. Technically, we use *superVP* in conjunction with the MuBu objects Schnell et al. (2009), to build this modular real-time synthesis engine of annotated audio samples. At each time step, gesture recognition is used to interactively select and time-stretch the audio segments according to the estimation of the temporal alignment on the reference. The segmental annotation of audio samples is used to design specific settings adapted to each type of gesture segments. Typically, the *Preparation* is linked to silence, *Attack* to a non-stretchable sound segment, *Sustain* to a stretchable sound segment, and *Release* to fading effect. Sustain segments are thus stretched or shortened whereas attack phases are played at the initial speed. In the specific case where the attack phase of the gesture is longer than that of the sound, the end of the gesture segment is time stretched to smooth the transition between audio processes.

A video that demonstrates the use of the system is available online.[9]

4.7 Summary and Contributions

We proposed a set of probabilistic models for movement modeling. We presented both instantaneous models with Gaussian Mixture Models (GMMs), and models that integrate temporal modeling, namely, Hidden Markov Models (HMMs) and Hierarchical Hidden Markov Models (HHMMs). Our implementation is grounded in an Interactive Machine Learning approach that stresses the importance of training from few examples and continuously performing recognition and analysis. We showed that parameters such as regularization give users the tools for designing expressive and efficient mapping strategies. Finally, we formalized for each model a set of mapping strategies for continuous sound control.

In particular, sequence models such as Hidden Markov Models (HMMs) allow us to derive temporal mapping strategies that account for motion and sound as continuous time processes. The development of higher level models such as the Hierarchical Hidden Markov Model (HHMM) pushes

9 http://vimeo.com/julesfrancoise/smc2012

further the integration of action-perception in the design process. As a matter of fact, it allows to specify sound control strategies based on a representation of motion and sound as a sequence of segments.

5

Analyzing (with) Probabilistic Models: A Use-Case in Tai Chi Performance

This chapters applies the probabilistic models introduced in Chapter 4 to continuous real-time movement analysis. We consider a use case in movement analysis where sequences of *T'ai chi ch'uan* movements (also referred as *Tai Chi* in this chapter) are executed by performers with varying expertise.

Along this chapter we analyze probabilistic models under two perspectives. First, we provide a methodology for performance analysis that draws upon probabilistic sequence models, with a focus on consistency in timing and dynamics Second, we aim to highlight the properties of the different models, to discuss their advantages and shortcomings, and to study the influence of their parameters.

5.1 Tai Chi Movement Dataset

We conducted a movement recording session at Ircam. We asked two participants to perform several trials of classical Tai Chi movement sequences. We focus on movements performed with the double-edge straight sword called *Jian*.

5.1.1 Tasks and Participants

We recruited two female performers: a dancer and professional Tai Chi teacher with several years of practice, and a Tai Chi student, trained but less experimented. In the subsequent sections, we refer to the teacher as participant **T**, and to the student as participant **S**.

Each participant was invited to execute ten performances of a long movement sequence containing approximately fifteen gestures for a total duration approaching forty-five seconds. The choice of this sequence was proposed by the participants who were used to practicing it. In order

to investigate interactive sonification of Tai Chi movements, we asked the teacher to vocalize along her performances (the cross-modal analysis is reported in Chapter 7). We did not ask the student to vocalize along her movements in order to avoid altering her performance with the supplementary task of producing vocal sounds.

5.1.2 Movement Capture

We recorded the movement of the performer and the vocal sound synchronously. All performances were videotaped, the vocalizations were recorded using a *DPA* microphone headset, and we captured performers' movements using inertial sensors. The performers were equipped with three mini-MO units (Rasamimanana et al., 2011), each containing a 3D accelerometer and a 3-axis gyroscope, yielding 18 data dimensions. The sensors were positioned as illustrated in Figure 5.1, with a MO on the handle of the sword, and the other two inertial units placed on the wrist and the upper arm, respectively.

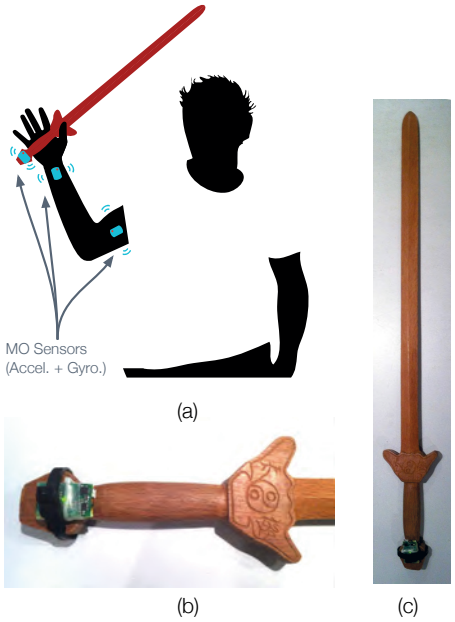


Figure 5.1: *Jian* sword Equipped with MO sensors, and disposition of other inertial sensors

5.1.3 Segmentation and Annotation

In the following sections, we compare three types of base segmentation for evaluating continuous recognition and alignment. These reference segmentations were created either by manual annotation or automatic segmentation:

MANUAL ANNOTATION was realized by the experimenters through the observation of the movement sequences. The segmentation is based

on key poses intersecting gesture "strokes", that often match minima or inflexion points of the global movement's energy. The segmentation contains either 12 or 13 segments depending on the trial[1].

REGULAR SEGMENTATION was performed by splitting the complete performance in a set of 12 segments with identical length.

MINIMUM-ENERGY SEGMENTATION looks for the 12 points of minimal energy — computed as the norm of the acceleration, — over the full performance, under the constraint of a minimal segment length of 1 second.

5.2 Analyzing with Probabilistic Models: A Methodology for HMM-based Movement Analysis

In this section, we study how the models' internal representation and parameters estimated from data can inform movement analysis. One of the great advantage of probabilistic models is their transparency. The parameters of Gaussian Mixtures Models and Hidden Markov Models have a clear interpretation that can enhance movement analysis methods. We illustrate how the interpretation of Markov models contributes to developing new methods in movement performance analysis.

Our primary topic of investigation is consistency. We aim to understand and interpret movement performances from people with different expertise. In this section we aim to illustrate how HMMs, and their hierarchical implementation, can provide two viewpoints on consistency as they simultaneously track temporal and dynamic variations.

5.2.1 Tracking temporal Variations

RELATED WORK Dynamic Time Warping (DTW) is widely used in movement analysis for its capacity to re-align movements in a non-uniform way, therefore allowing to compare performances with complex timing variations.

For example, Ferguson et al. (2014) recently conducted a study where DTW was used to assess the changes in timing between several performances of a dance sequence under different conditions (music vs non-music). The authors derive two measures of scaling and lapsing to identify long- and short-term timing variations.

DTW suffers from a quadratic time and space complexity, as it draws upon Dynamic Programming that requires the full observation sequences for computing the alignment path. Implementing DTW in real-time remains difficult even though alternative implementations such as the *LB_Keogh* (Keogh and Ratanamahatana, 2004) method address complexity issues. Recently, Gillian Gillian et al. (2011) proposed a real-time implementation of DTW for multidimensional gestures that focused mostly on recognition rather than warping analysis.

1 See section 5.4.1 for details.

Alternatively, Bevilacqua et al. (2011, 2010) proposed a template-based implementation of HMMs called *Gesture Follower* that allows for real-time sequence warping. HMMs are built from a single example by associating a state to each sample of the template gesture in a left-right transition structure. The warping path can be computed in real-time using a forward algorithm: the forward variable allows to recover the *time progression* within the template recording.

HMM-BASED TIME WARPING

We propose to extend the approach of Bevilacqua et al. (2010) to generic HMMs that can be learned from multiple performances. We presented in Section 4.4.1 a method for estimating the normalized time progression within the model based on the expectation of the distribution of state posteriors. Sensible choices of the model's parameters — a large number of states and intermediate regularization, — makes it possible to compute a smooth and continuous alignment. The alignment can be used to derive a warping path to realign the performances. The main limitation of the method is the smoothness of the alignment path: while using few states reduces the complexity of real-time inference, it also degrades the accuracy of the estimated time progression which tends to a step function when states scarcely overlap.

HIERARCHICAL IMPLEMENTATION

Although it is possible to compute the alignment of the full sequence using a single model — if gesture following is the only focus, — this task is severely limited by the complexity of the training algorithm. When using HMMs for gesture following, a large number of hidden states is required to guarantee a sufficient temporal accuracy, and training can become too long for an interactive setting.

We propose an alternative solution that uses a HHMM to implement a divide and conquer approach to the alignment problem. In this case, we use a reference segmentation to define a sequence of consecutive motion segments, that are used to train a hierarchical HMM. To compute the warping path, we evaluate the time progression as the time progression of the likeliest segment, which allows to reconstruct the index of the warping path. In the remainder of this section, we use a Hierarchical HMM with a left-right topology, trained with examples segmented with the manual annotation.

ILLUSTRATIVE EXAMPLE

Figure 5.2 shows the first 20 seconds of the realignment of two trials of participant **T** over the full sequence. The figure depicts the raw sensor values of performances 2 and 3, and the realignement of performance 2 to performance 3 using the HHMM method and DTW. A HHMM was trained with trial 2, and trial 3 was used for testing. We used a single Gaussian and 50 states per segment, raising a total of 650 states; the absolute regularization was set to $\lfloor\sigma\rfloor = 0.01$, no relative regularization was used.

The estimated time progression was used to reindex the training example to fit the timing of the test performance. We performed a similar operation with multidimensional DTW using the euclidean distance. In both

cases, we used a 51-long moving average filter to smooth the warping path and avoid signal variations due to random noise to be warped too strictly.

The realigned performance using the HMM-based method is very close to one obtained with DTW, which validates our method as a good approximation of DTW for sequence warping. Moreover, regularization can be used as a warping constraint. Using small regularization increases the dependency over the observation model and provides a strong alignment that will warp all variations in the signal. At the opposite, a large offset on the variances tends to increase the contribution of the transition structure and relaxes the dependency on signal variations, thus providing a smoother temporal stretching. Most importantly, the alignment is performed in real-time using the forward algorithm, and therefore can be used for continuous interaction.

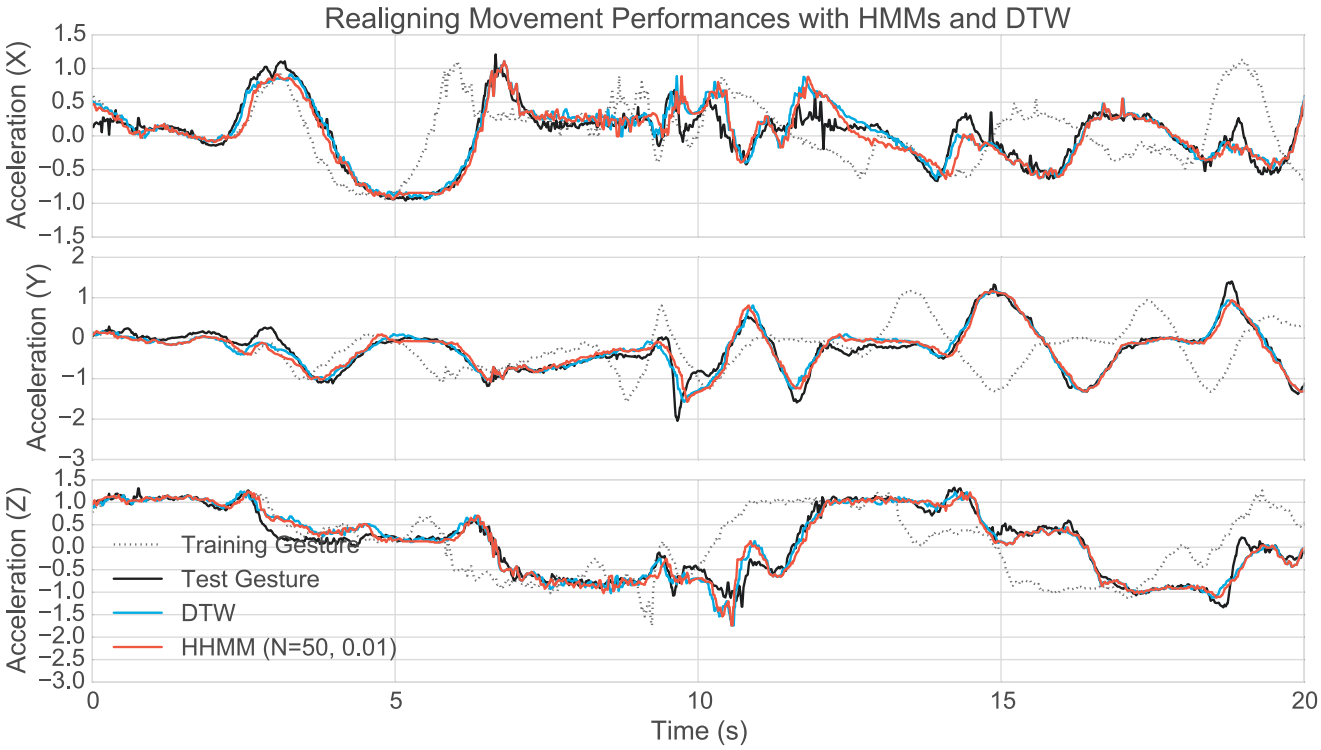


Figure 5.2: Alignment of performances with HMM and DTW. A HMM is trained using trial 2, and trial 3 is used for testing. The warping paths computed with HMMs and DTW are used to re-align the training gesture to the test gesture. The HHMM was trained with 50 states per segment (650 total states), a single Gaussian and regularization $\lfloor\sigma\rfloor = 0.01$.

5.2.2 Tracking Dynamic Variations

While computing the warping path informs on the timing variations between performances, it does not account for changes in dynamics. As illustrated in Figure 5.2, superimposing the re-aligned performances provides visual insights into the dynamic changes between performances. For ex-

ample, while the warped performance in the example is fairly close to the reference over the full performance, we can observe that the gestures performed around 10 seconds have very different dynamics in the two performances. One way to quantitatively assess these dynamic changes is to measure the Root Mean Square (RMS) error between motion parameters on the aligned movements.

The interest of the HMM framework is that it provides a joint measure of the alignment and dynamic costs through the likelihood: both changes in timing and changes in dynamics impact the likelihood of the sequence given a model.

ILLUSTRATIVE EXAMPLE Figure 5.3 illustrates the parameters that can be extracted in real-time with HMM-based continuous movement following. These parameters provide valuable insights into performance analysis. On this example, recorded by participant **T**, we can observe from the time deviation that, compared to performance 2, the participant progressively accelerates between 3 and 4 seconds, than presents to sharper accelerations around 5 and 7 seconds. A lot of variations occur between 9 and 13 seconds, that relate both to timing and dynamics, as illustrated through the RMS error and the log-likelihood.

Interestingly, the log-likelihood strongly correlates with the RMS error rather than the temporal variations. Actually, the cost for temporal warping is usually very inferior to the cost of dynamic changes. The left-right transition structure allows flexible timing changes while the observation models are very discriminant and therefore more sensitive to changes in dynamics. The log-likelihood provides a more consistent and smoother measure of the changes in dynamics than the RMS error, and does not require additional computation.

5.2.3 Discussion

Hidden Markov Models (HMMs) provide a flexible tool for online performance analysis. With appropriate parameters, HMMs can be used for performance warping similarly to Dynamic Time Warping (DTW). The advantage of using HMMs are threefold. First, they allow for online alignment whereas DTW is more computationally expensive and requires the entire motion sequence. Second, regularization allows to define smoothness constraints on the warping path. Third, HMMs can be trained with several examples, and the warping can therefore be made to an average performance, taking into account the variability across several trials. A disadvantage of the HMM framework is that tuning the parameters might be tedious in comparison with Dynamic Time Warping (DTW) that does not require additional parameters.

Finally, the likelihood gives a measure of the similarity of a performance's dynamics in comparison with a reference model, that allows us to investigate how the consistency of a performer evolves over time. However, log-likelihoods are highly dependent on the training examples as well

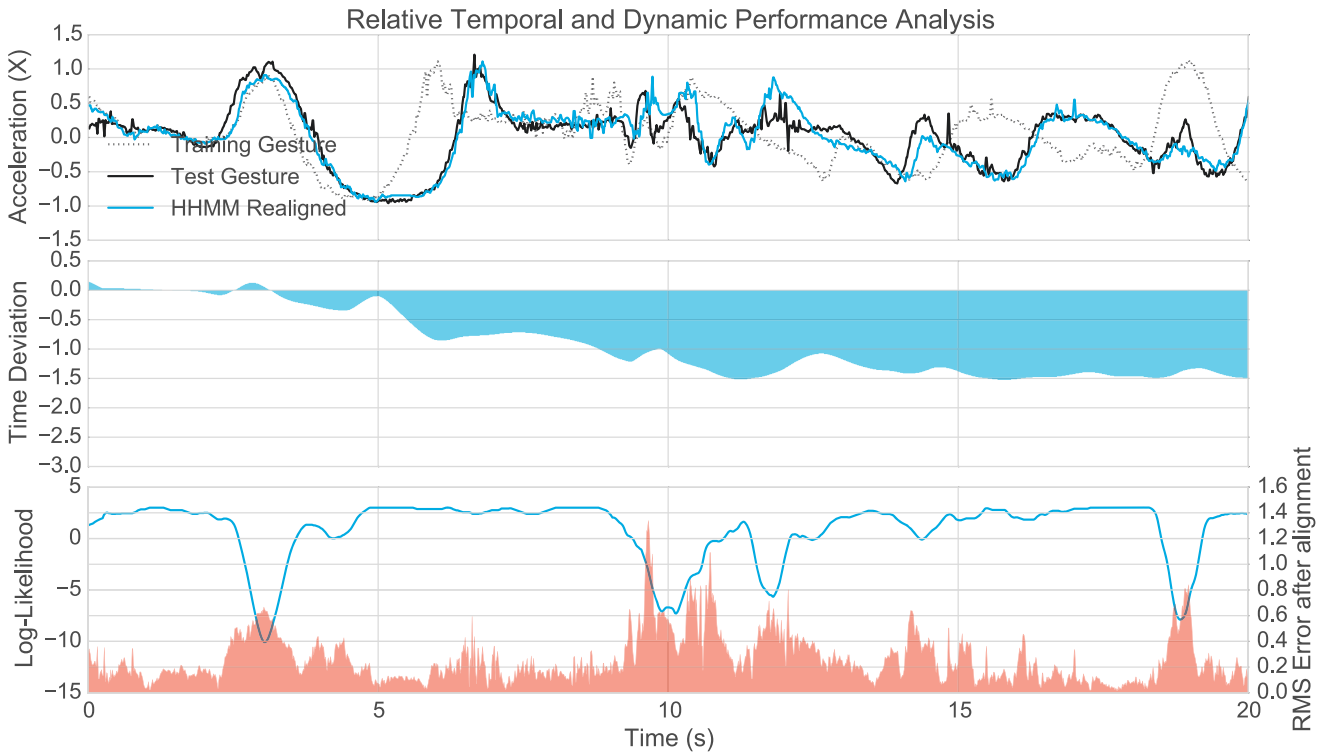


Figure 5.3: Timing and dynamics analysis with HMMs. The figure analyses the differences between two performances through the re-alignment (blue) of trial 2 (dotted black) over trial 3 (black). Time deviation is calculated as the difference between the time progression and the true time, the RMS error is measured over the three axes of the acceleration on the re-aligned performances, and the likelihood is averaged over a window of 51 samples. The HHMM was trained with 50 states per segment (650 total states), a single Gaussian and regularization $\lfloor\sigma\rfloor = 0.01$.

as on the parameters, and do not provide per se an 'absolute' measure of consistency.

Both the time progression and the log-likelihood can be computed in real-time, continuously, and can therefore serve as control parameters for the design of continuous interactions.

5.3 Evaluating Continuous Gesture Recognition and Alignment

In the following sections, we detail the results of a joint segmentation, recognition and alignment task where we aim at segmenting and following a performance in real-time, based on a model learned on one or more recordings of the same movement sequence. We propose to compare the different models and evaluate the influence of their parameters using the following general protocol.

5.3.1 Protocol

First, we extract a set of segments from one or several trials out of the ten recordings of each participants, either using automatic segmentation or using a manual annotation. Then, we evaluate the recognition and alignment computed with the different models on all remaining trials. This way, we guarantee that the training examples are not used for testing, but we generate enough combination to get sufficient statistics. For each test sequence, we compute in a real-time setting the sequence of recognized labels $\{c_t^{(rec)}\}_{t=1:T}$ and temporal alignment $\{\tau_t^{(rec)}\}_{t=1:T}$ as detailed in Section 4.2.2 and 4.6.1 as

$$c_t^{(rec)} = \underset{c}{\operatorname{argmax}}\, p(c \mid \boldsymbol{x}_{1:t}) \tag{5.1a}$$

$$\tau_t^{(rec)} = A(c) + B(c)\bar{\tau}(\boldsymbol{x}_{1:t} \mid c) \tag{5.1b}$$

where $A(c)$ and $B(c)$ are respectively the starting time and duration of segment c, and where $\bar{\tau}$ is the normalized time progression.

5.3.2 Evaluation Metrics

We propose to jointly evaluate segmentation and recognition for the purpose of continuous interaction. Our goal is not to compute *a posteriori* a sequence of the identified segments with their errors and delays, but rather to continuously recognize, label, and follow the gestures.

The proposed models and inference algorithms compute, at each new observation, both the likelihood of each segment and the temporal alignment within the segment. We propose a metric for evaluating continuous recognition that expresses the proportion of time where the segments are correctly identified. Consider a movement sequence of length T, and its associated sequence of true labels $\{c_t^{(true)}\}_{t=1:T}$ and recognized labels $\{c_t^{(rec)}\}_{t=1:T}$, we define the *recognition error* as

$$\epsilon_{seg} = \frac{1}{T}\sum_{t=1}^{T}\left[1 - \delta(c_t^{(true)}, c_t^{(rec)})\right] \tag{5.2}$$

We propose a similar metric to evaluate continuous alignment. In this case, we aim to measure the distance between the timing of the gesture and the temporal alignment evaluated by the algorithm. Let $\{\tau_t^{(true)}\}_{t=1:T}$ be the true time progression in seconds and $\{\tau_t^{(rec)}\}_{t=1:T}$ be the predicted time progression. We define the *alignment error* as

$$\epsilon_{align} = \frac{1}{T}\sum_{t=1}^{T}\left|\tau_t^{(true)} - \tau_t^{(rec)}\right| \tag{5.3}$$

5.3.3 Compared Models

We propose to compare the results of joint real-time segmentation and recognition with Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), and the Hierarchical Hidden Markov Model (HHMM). We evaluate two topologies for the top level of the Hierarchical HMM:

ERGODIC The ergodic structure allows all possible transitions between motion segments with equal probability

LEFT-RIGHT We built a strict left-right structure which only allows transitions between segments in the temporal order of the sequence. The high level transition matrix takes the form of a first diagonal matrix.

The next sections are organized as follows. We start by comparing the four probabilistic models for continuous real-time gesture segmentation, recognition, alignment, and spotting. We evaluate the results and discuss the type of errors for each model to identify how to *get the right model* for dynamic movement recognition. Then, we present detailed results of the influence of the models' parameters — e.g. the number of states or regularization, — and discuss their interpretation. This second section aims to guide the design of models for movement modeling, and relates to *getting the model right.*

5.4 Getting the Right Model: Comparing Models for Recognition

In the following results, we evaluate the segmentation of a long sequence (about 45 seconds) with a model trained on a single example of the same sequence performed by the same participant. With ten trials available, and keeping an example for training, we compute the segmentation and recognition on the nine remaining trials, raising 90 segmentations for each statistics — note that recognition errors are averaged within each fold of the training set. In this section, we consider only *manual annotation* as the reference segmentation for training and evaluation.

5.4.1 Continuous Recognition

TEMPORAL MODELING The left part of Figure 5.4 reports the recognition error for participant **T** across all performances for the four probabilistic models. All models are trained with a single Gaussian, and we use 10 states for Markov models. The regularization was set to its average optimal value across models — the influence of this parameter is further explained and evaluated in Section 5.5.

The HHMM reaches 9.4% and 10.4% recognition error with ergodic and left-right topologies, while HMMs achieve 18.0% error, and GMMs 30.0% error. An Analysis Of Variance (ANOVA) yielded significant differences between the models ($F_{3,36} = 196.9$, $p < w > 0.001$), and a post-hoc analysis with the Tukey-Kramer method showed that the HHMM performs significantly better than HMMs, themselves performing significantly better than GMMs at $p < 0.001$. However, no significant difference was found between the two topologies for the HHMM at $p < 0.05$.

These results confirm the need for time series models for analyzing expert movement. Indeed, the inability of GMMs to account for temporal modeling results, as expected, in decreased performance on movement segmentation. Among the Hidden Markov models and extensions, the hi-

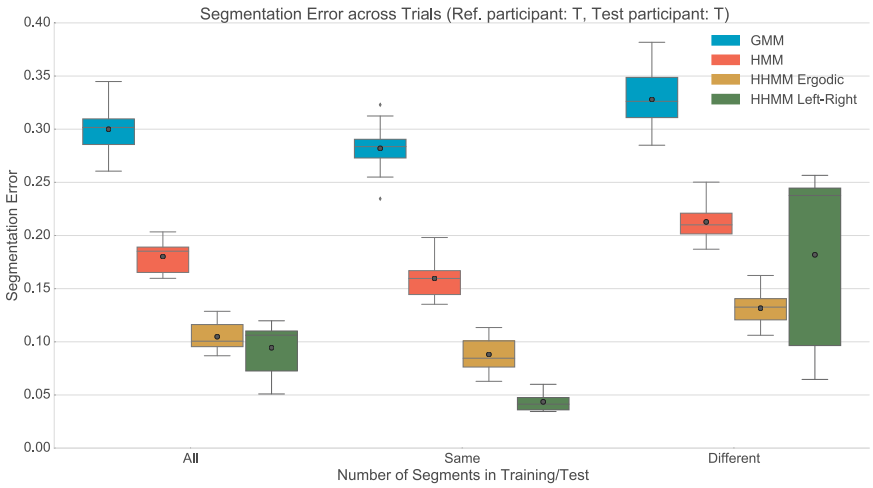


Figure 5.4: Box plot of the recognition error across all trials for participant **T**, compared by the consistency of the number of segments in the training and test trials. Each box represents the first, second and third quartiles. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 1e^{-2}$, $L_W = 1$).

erarchical structure of the HHMM proves to perform significantly better than a set of HMMs running in parallel. This observation supports the argument that the addition of a high-level structure governing transitions between unitary segments is crucial to improve real-time segmentation and recognition.

PRIOR KNOWLEDGE AND CONSISTENCY

It can be surprising, however, that adding a prior on the transitions between segments degrades the performance on segmentation. Here, we chose a left-right transition structure that only allows transitions to the next segment, as the movement sequences were always performed in the same order. We observe that the variability of the recognition error is larger than with the ergodic structure: the distribution of recognition errors presents outliers, which increases the mean recognition errors to the third quartile. This variability can be explained through the observation of the various trials. As a matter of fact, some trials are performed with a variation: three performances contain one additional segment, repetition of a short gesture. Therefore, three trials contain 13 segments instead of 12, which explains why the forced transition structure fails at improving the recognition.

To further highlight this issue, we present the results according to the consistency of the number of segments in the training and test performances. The filtered results for the trials that have the same number of segments in training and testing are plotted in the center box plot of Figure 5.4. In this case the statistics are computed over 48 segmentations.

Unsurprisingly, all models perform better than when averaging across all trials: GMMs and HMMs respectively drop to 28.2% and 16.0% recognition error while hierarchical HMMs reach 8.8% and 4.3% for the ergodic and left-

right structure. An ANOVA confirmed that this difference of performance is significant for each model ($F_{2,36} = 317$, $p < 0.001$). Most importantly, the HHMM with a left-right transition structure now performs significantly better than with an ergodic structure. An ANOVA highlighted significant differences between the models, and a post-hoc analysis with the Tukey-Kramer method at $p < 0.001$ showed a significant difference for all pairwise model comparisons.

This result confirms that adding prior information helps improving segmentation when having a high degree of certainty on the scheduling of segments. Indeed, when higher variability arises, adding such a strong prior tends to propagate errors more dramatically than with a moderate prior.

> Sequence models such as HMMs outperform instantaneous models (i.e. GMMs), because of the temporal modeling that is necessary for encoding dynamic gestures. The Hierarchical HMM adds a high level structure that makes the model more discriminant on continuous recognition. Adding prior knowledge to the sequencing of the motion segments helps improving the recognition but can lead to critical errors when the prior is too strict with regards to the performer's consistency.

5.4.2 Types of Errors

ANALYSIS OF SEGMENT LENGTHS

We presented global statistics on recognition error that only account for the ratio of time the models correctly label the current frame. However, this measure is limited in that it poorly represents the type of recognition errors that arise. Figure 5.5 depicts the histograms of the recognized segment lengths — computed as the number of consecutive frames with the same label.

While the segment lengths for the hierarchical models are distributed closely to the reference segmentation, GMMS and HMMs present a lot more short segments about a few samples long. In practice, this distributions show that GMMs and HMMs tend to switch quickly between several labels, whereas the hierarchical HMMs have a more stable behavior resulting in longer segments.

This consideration highlights the different types of errors that occur in joint real-time segmentation and recognition: while true classification errors imply that an entire motion segment is wrongly labeled, *insertions* occur when the recognition quickly switches between different labels — therefore inserting a short error within a correctly recognized motion segment.

EXAMPLE

Figure 5.6 illustrates an example of the segmentations obtained with the four models. It appears that for GMMs and HMMs, the recognition often 'jumps' quickly between classes. On the other hand, the hierarchical model have a more regular response — especially when using a left-right high level transition structure, — that implies that errors

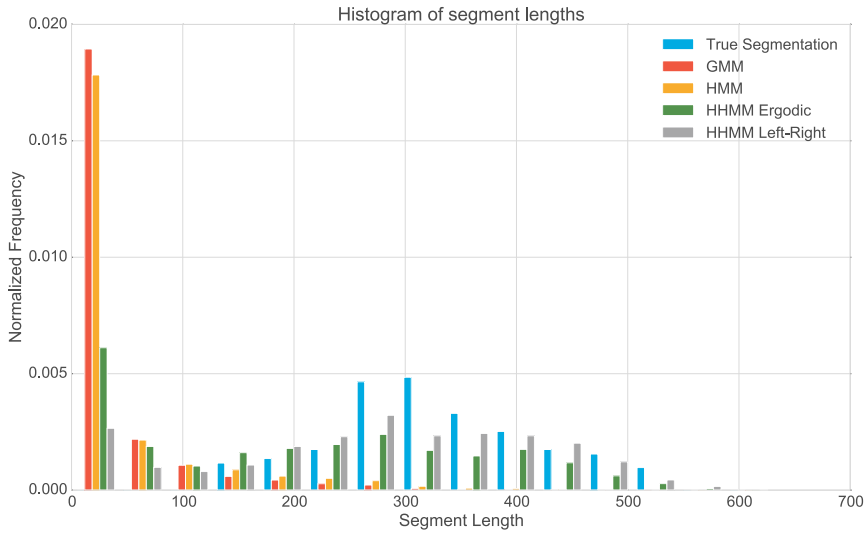


Figure 5.5: Histogram of the lengths of recognized segments for participant **T**, compared by model and base segmentation. Segment lengths are computed as the number of consecutive frames with the same label. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 1e^{-2}$, $L_W = 1$).

are mostly due to temporal shifts or delays in the recognition rather than classification errors. This assessment is crucial for usability in interaction: when using continuous recognition, it is important to ensure both a low error rate and a stable response of the system.

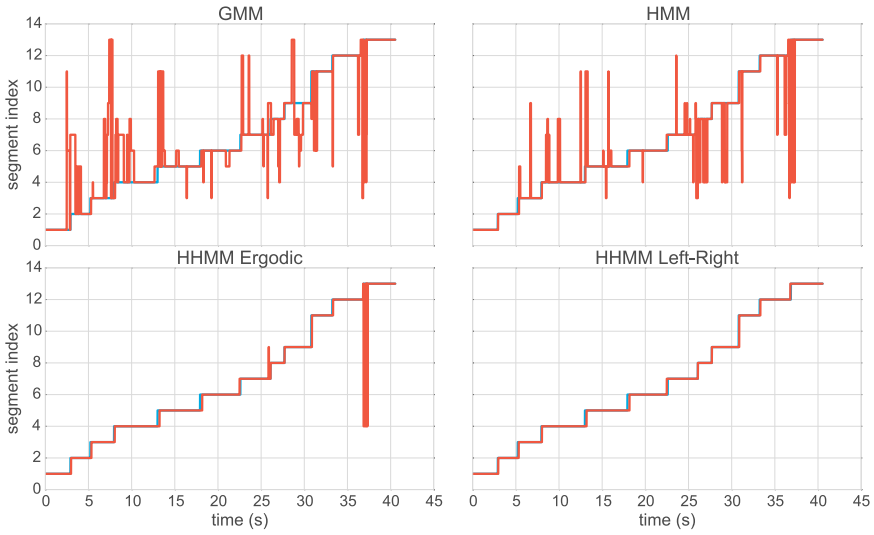


Figure 5.6: Example of segmentation results from the four models for participant **T**. The blue line represents the true segmentation, the red line the segmentation computed in real-time. Segments are indexed from 1 to 13 in their order of appearance in the sequence. The models were trained on trial 2 with 10 states and a single Gaussians ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 1e^{-2}$, $L_W = 1$). Trial 3 is used for testing.

GMMs and HMMs fail at capturing the temporal dependencies and results in unstable recognition patterns. Hierarchical models ensure both a more stable response and a lower error rate, which is highly desirable for continuous interaction.

5.4.3 Comparing Models for Continuous Alignment

We proposed another metric aiming to evaluate the quality of real-time alignment to the reference gesture. The alignment error measures the euclidean distance between the time progression computed with a particular model and the true time progression.

Figure 5.7 details the alignment error for the three temporal models, according to the equality of the number of segments in the training and test performances. The results are very similar to the recognition error: for identical number of segments in training and testing, an ANOVA yielded significant differences between models ($F_{3,27} = 93$, $p < 0.001$), and a post-hoc analysis with the Tukey-Kramer method under $p < 0.001$ showed significant differences between all models in the following order of decreasing alignment error: HMMs, Ergodic HHMM, Left-Right HHMM. The results obtained with the hierarchical model confirm that the prior on segment sequencing significantly improves the alignment when the performer is consistent enough to guarantee the respect of the long-term temporal structure.

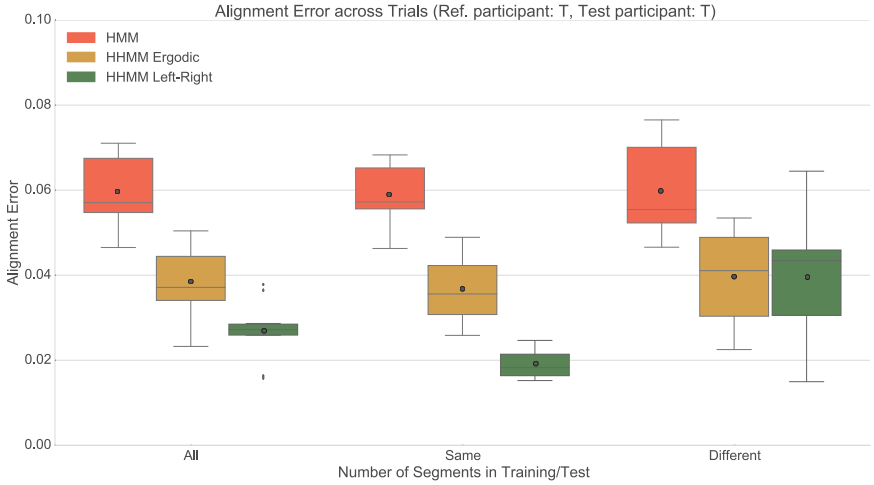


Figure 5.7: Box plot of the alignment error across all trials for participant **T**, compared by the consistency of the number of segments in training and test performances. Each box represents the first, second and third quartiles. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 1e^{-2}$, $L_W = 1$).

If we consider the alignment error on the frames that are correctly labeled — therefore discarding recognition errors, — we observe that all models have a similar performance on alignment. As all of the Markov models have the same internal representation of motion segments, they show the same ability to compute real-time alignment. Therefore, align-

ment errors are mostly due to recognition errors. This result underlines that the most critical problem for real-time gesture following is to correctly segment and label motion segments.

5.4.4 Gesture Spotting: Models and Strategies

We presented results on real-time gesture segmentation where the task was to recognize and follow a sequence of consecutive motion segments. The movement sequence continuously chained a set of identified gestures, and the task therefore amounted to classifying segments one against each other. However, in most real-world problems, continuous gesture recognition is not only limited to correctly labeling gestures in real-time. It must also identify from a continuous stream of movement parameters the temporal boundaries between "meaningful" gestures and movements that are not aimed to be recognized — thereafter called "filler" movements.

GESTURE SPOTTING We now address this problem of gesture *spotting*, which aims to identify, recognize and label specific gestures in a continuous stream of movement. For this purpose, we use the same recordings of Tai Chi movements, but we perform the recognition task on a subset of the gesture segments.

The protocol for training and recognition was modified as follows. For each performance used for training, we randomly select S segments to spot among the 12 segments common to all performances. All other segments are labeled 0, and represent "filler" movement. We perform the recognition on all other trials with the procedure described in Section 5.3.

SPOTTING STRATEGIES We presented in Section 4.2.2 two strategies for gesture spotting. The first is based on thresholding the log-likelihood. An alternative strategy consists in training a *filler* model on all instances of "filler" movement, that we call *model-based* spotting in the following. In this section we propose to compare these strategies for the 3 proposed models: GMMs, HMMs and the HHMM.

For likelihood-based spotting, we train only the gesture segments to recognize, and we define the threshold using the recognition of the training example itself. In the following evaluation, we use as threshold either the minimum or the 5% percentile of the log-likelihood on the portions of movement to recognize.

For evaluation we use the same metrics of recognition and recognition error with the new labelization, and therefore account for three types of errors: gesture segments recognized on filler portions (Type 1), gesture segments not recognized (Type 2), and incorrectly labeled gesture segments (Type 3).

RESULTS The results of the spotting tasks are presented in Figure 5.8. The figure compares the recognition error obtained with each of the two proposed strategies: model-based spotting and likelihood-based spot-

ting, for several numbers of segments to spot among the 12 total segments.

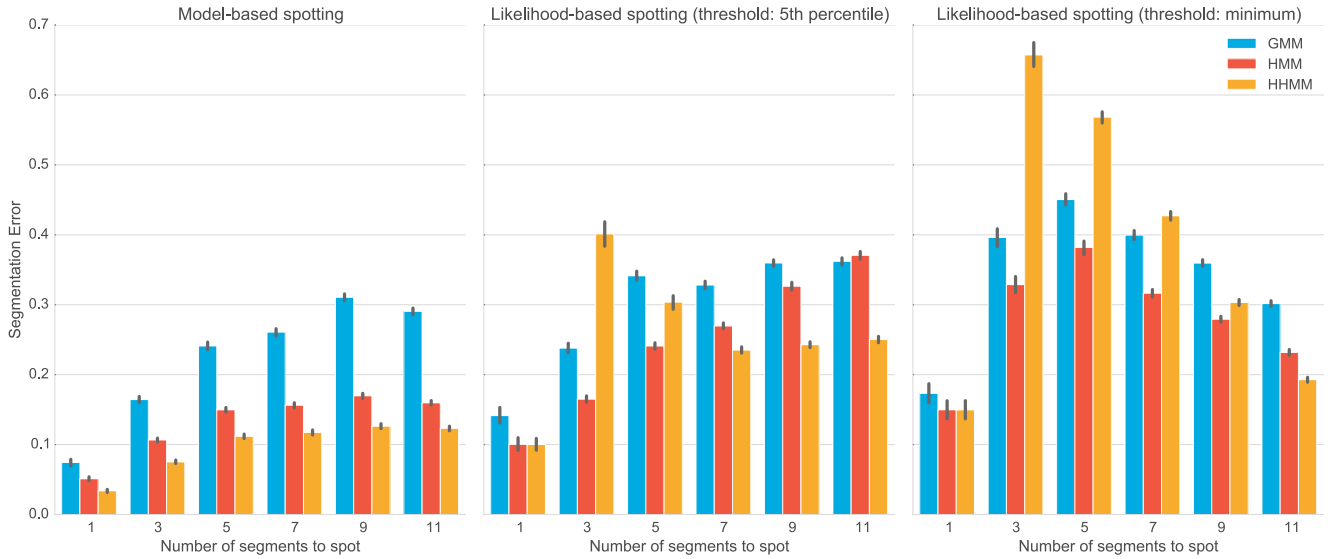


Figure 5.8: recognition error for spotting across all trials for participant **T**, compared by number of segments to spot and spotting strategy. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $L_W = 1$). We used two different values of regularization for model-based spotting ($\lfloor\sigma\rfloor = 1e^{-2}$) and likelihood-based spotting ($\lfloor\sigma\rfloor = 5e^{-2}$) to optimize each method.

The model-based spotting strategy clearly outperforms the approach based on likelihood thresholding, whatever the threshold used for the log-likelihood. As a matter of fact, the model-based approach presents a significantly lower recognition error for all numbers of segments to spot. We observe that the recognition error increases with the number of segments to recognize, because recognition errors (Type 3) naturally increase with the number of classes.

However, for likelihood-based spotting, the results can vary with the chosen log-likelihood threshold: a high threshold presents an increasing error with the number of segments to recognize, while a lower threshold performs poorly with few segments but better with numerous segments.

TYPES OF ERRORS Analyzing the types of errors gives insights into each strategy. The repartition of the types of errors across trials is presented in Figure 5.9.

For model-based spotting, most of the errors occurring for low number of segments are due to false negatives (missed segments), while the proportion of classification errors is higher with numerous segments to recognize. Likelihood-based spotting with a high threshold presents a similar behavior except that the performance degrades as the number of segments increases: many false positives, the model becomes less discriminant. On the other hand, using a lower threshold tends to favor false positives, which

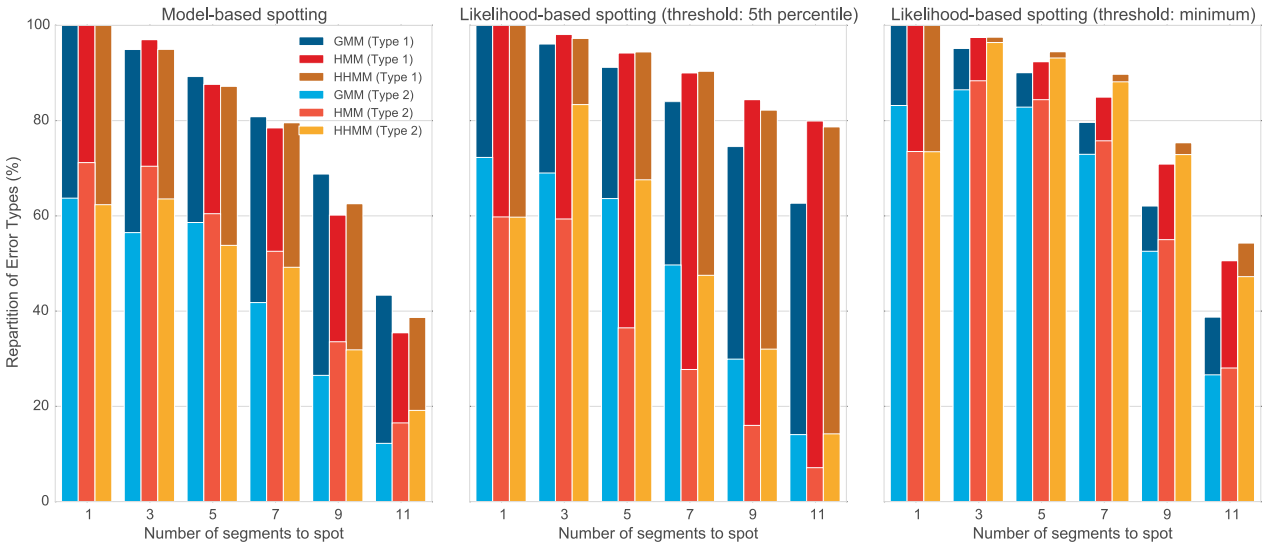


Figure 5.9: Repartition of spotting errors all trials for participant **T**, compared by number of segments to spot and spotting strategy. Type 1 error occur when a segment is labeled in place of non-gesture (false positive), Type 2 errors occur when a segment is missed (false negative), and the remaining are classification errors. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 5e^{-2}$, $L_W = 1$).

degrades the performance for few segments but decreases the recognition errors when many classes are to be recognized.

DISCUSSION The likelihood-based approach therefore presents the advantage of providing a possible compromise between the types of errors. Depending on the context of use, one might prefer a strict policy on spotting (avoiding false positives) while other use cases might favor false negatives to ensure that all gestures are identified. However, two drawbacks limit the usability of the approach: tuning the threshold might be tedious and time consuming, and the accuracy on spotting might always be inferior to model-based spotting. On the other hand, model-based spotting presents the advantage of having no additional parameter to tune, but requires additional data of "filler" movement to be efficient. In our case, the model-based approach performs especially well, because the filler movement is consistent across trials. A more extensive study with several participants might be required to confirm these results.

5.5 Getting the Model Right: Analyzing Probabilistic Models' Parameters

The previous section focused on comparing together the proposed probabilistic models, i.e. *getting the right model*. In this section, we focus on

getting the model right: understanding and optimizing the models' parameters depending on the aims of the task and the context of use.

5.5.1 Types of Reference Segmentation

We now address the issue of comparing the performance of the models on different types of base segmentation. As discussed in Section 5.2.1, we gain in modularity by using segmentations of long performances, and we also gain significantly in algorithmic complexity — particularly for the training algorithm.

We evaluated and compared the different models using a manual annotation. However, such segmentation can be tedious and requires both time and expert knowledge. In order to assess the models' ability to perform segmentation and alignment in other contexts, we now evaluate manual annotation with respect to two automatic segmentation methods: a regular segmentation and a segmentation based on minima of the acceleration energy.

RESULTS The alignment error across trials for each model and each type of base segmentation are plotted in Figure 5.10. We observe almost no different between the three types of segmentation for movement alignment. While the regular and energy-based segmentation give poor recognition results, due to the inconsistency of their labeling, such ambiguities do not impact the ability to predict the time progression.

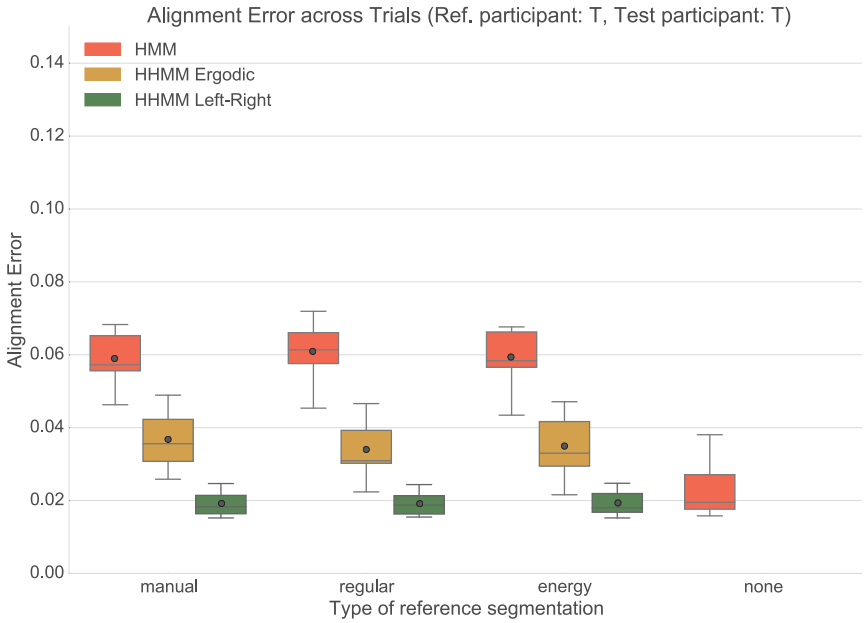


Figure 5.10: Box plot of the alignment error across all trials for participant **T** for the three base segmentations. Each box represents the first, second and third quartiles. Models are trained with a single Gaussian component and 10 hidden states ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = 5e^{-2}$, $L_W = 1$).

We also compared with the a baseline situation where we don't use any segmentation. We train a single HMM with 130 states, in order to ensure the same temporal accuracy, and perform a measure of alignment. Interestingly, if it performs better than HMMs and the ergodic version of the HHMM, no significant difference in alignment error was found with the left-right hierarchical model, whatever the base segmentation.

ACCURACY OF THE DIVIDE AND CONQUER STRATEGY

Using a hierarchical model with an arbitrary segmentation yields the same accuracy as a flat model for real-time gesture following. The major difference lies in the complexity of the training algorithm. Indeed, the hierarchical HMM intrinsically divides the training algorithm in the training of submodels that have a lower number of states and are trained on shorter observation sequences. This process critically reduces the time and memory complexity of the Baum-Welch algorithm.

Although approximations and advanced optimization methods could be used to derive a faster implementation of large HMMs — such as the windowing technique used in Gesture Follower, — the hierarchical structure guarantees exact inference and allows for parallel training of the submodels.[2]

> Using a Hierarchical HMM with an arbitrary segmentation yields the same accuracy as a flat HMM model for real-time gesture following. As a results, the hierarchical approach provides a divide and conquer strategies for continuous alignment, that is particularly advantageous when working with a large number of states.

5.5.2 Model Complexity: Number of Hidden States and Gaussian Components

The number of hidden states for HMMs and extensions and the number of Gaussian components for both GMMs and HMMs specifies the desired temporal accuracy or non-linearity of the model. In order to assess the influence of this parameter on the joint segmentation and recognition task, we ran the segmentation for different values of the number of hidden states and Gaussian Components. The results on real-time recognition are presented in Figure 5.11 for participant **T**.

We first consider the case with GMMs for various values of the number of Gaussian components. While it seems natural that using a single Gaussian clearly tends to underfitting, it is more surprising that the best recognition score is reached for 2 components, while the segmentation error is higher with 5 components. Because of the independence assumption between observations, increasing the number of components of the GMMs soon results in overfitting: if several segments in the sequence contain similar poses (or similar sensor values), they might be easily confused if they are not embedded in a consistent temporal modeling framework.

On the contrary, all sequence models — HMMs and Hierarchical HMMs, — present a consistent behavior where the segmentation error decreases

2 In our implementation, we use multithreading to train several classes in parallel

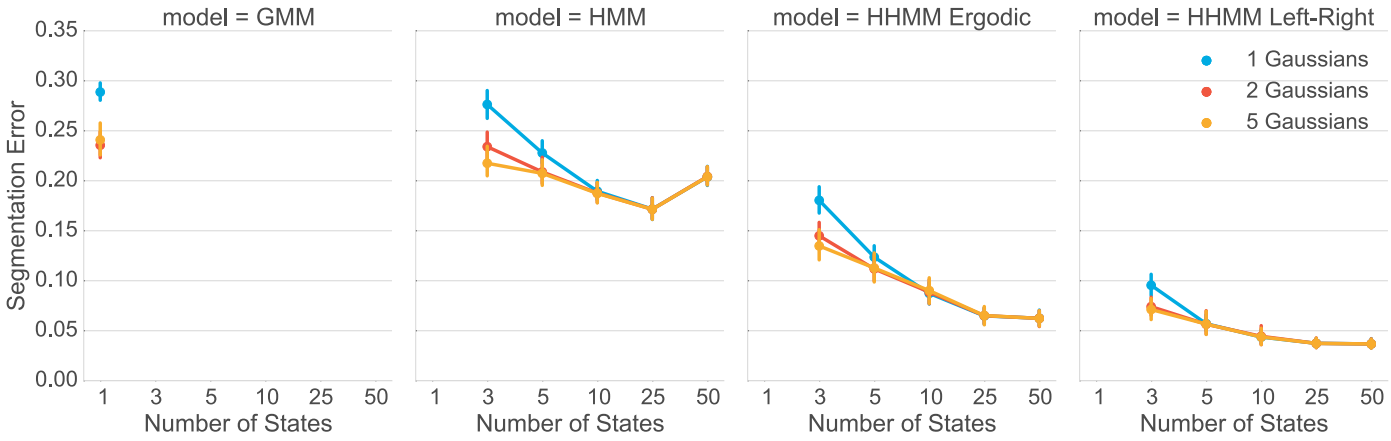


Figure 5.11: Influence of the number of hidden states and Gaussian mixtures on gesture segmentation (Teacher)

as the number of hidden states increases. This observation is true up to 25 hidden states. Using more states (e.g. 50 states) can result in overfitting: the performance of both HMMs and HHMM with a left-right structure decreases after 25 states.

5.5.3 Regularization

We proposed in Sections 4.1.4 and 4.3.5 to regularize the variances of the Gaussian components to provide a spatial smoothing strategy that artificially increases the generalization of a model trained on few examples.

In this section we investigate the impact of regularization on real-time recognition and alignment. Using the same evaluation procedure, we tested for each model –trained with 10 states and a single Gaussian, — several levels of regularization.

Figure 5.12 details the segmentation error for values of the relative regularization — that is proportional to the variance of each data channel over the training set, — spanning from $1e^{-4}$ to 10.

Several observations can be made from the results of the recognition. First, there is a critical minimal value of the regularization that guarantees consistent results. For the case of participant **T**, regularizing below $1e^{-2}$ results in very poor performance of the four models. In this case, the values are too small to guarantee a consistent training. We chose a critical situation in terms of learning where we use a single training example to estimate all of the HMMs' parameters. The training set is therefore too small to yield a consistent estimate of the variances, which results in a poor generalization of the models. At the other side of the spectrum, large regularization values (above 0.5 in this example) can also degrade the models' performance on real-time recognition. In this case, the variance prior is so large that it blurs the recognition boundaries: each model becoming too general, the recognition process looses its discriminative power. In our example, the optimal range of values of the regularization spans between

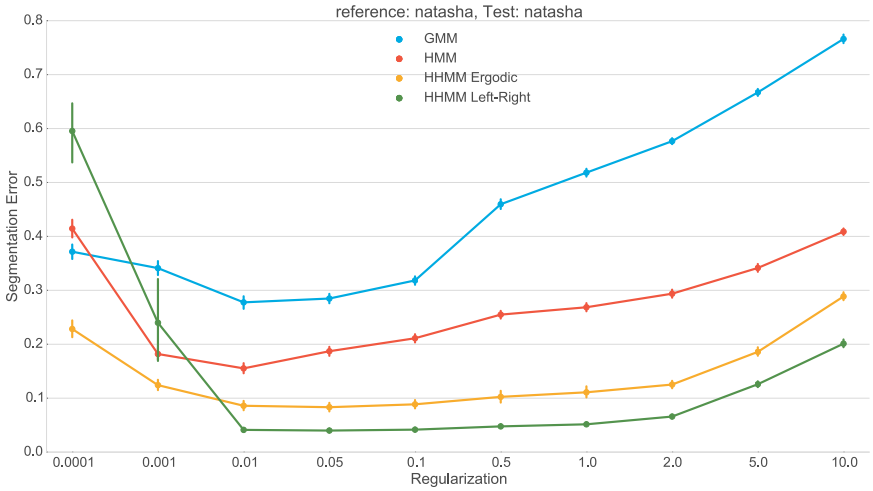


Figure 5.12: Influence of the regularization on gesture segmentation

$1e^{-2}$ and $1e^{-1}$. This result is consistent with general observations and experience of using the models for recognition and interactive sonification, and might be an appropriate choice in first approximation for most applications in continuous movement interaction.

> Regularization is a crucial parameter for continuous recognition when the models are trained on few examples, as it allows to avoid overfitting.

5.6 Summary and Contributions

In this section we applied the probabilistic models presented in Chapter 4 to movement analysis. We showed how Hidden Markov Models (HMMs) could be used for online continuous analysis of performance timing and dynamics. We proposed a method for sequence alignment based on a divide-and-conquer method with the Hierarchical Hidden Markov Model (HHMM) that critically reduces the training complexity.

We compared four probabilistic movement models on a joint segmentation, recognition and alignment task: Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and Hierarchical Hidden Markov Model (HHMM) with two topologies: ergodic and left-right transition structures. The results of the joint segmentation and recognition task emphasize the importance of using sequence models for recognizing dynamic movement accurately. Adding high-level prior information improves the recognition when the performance respect the sequencing constraints. We showed that the number of hidden states, regularization and the temporal smoothing strategies give users transparent parameters for optimizing gesture recognition and alignment.

6

Probabilistic Models for Sound Parameter Generation

In chapter 4, we presented several strategies for designing sonic interactions based on probabilistic models of movement. With movement models, most of the interaction techniques emerge from the recognition process. Often, we create an analytical mapping from the recognized gesture's label, likelihood, and time progression to the control parameters of a synthesis model.

GENERAL APPROACH In this chapter, we detail how probabilistic models can be used for a more integrated control of sound synthesis through parameter generation algorithms. Drawing upon multimodal representations of movement and sound data, we develop several strategies to encode the cross-modal relationships between movement parameters and audio processing.

Our approach to probabilistic mapping of movement and sound can be summarize as follows. First, we learn a joint model of motion and sound, by estimating a distribution over the joint feature space composed by motion and sound parameters. Then, we convert the joint model to a conditional model, that expresses the distribution over sound parameters conditionally to the distribution of motion parameters. This conditional model allows us to perform statistical parametric synthesis: the model generates sound parameter sequences given motion parameter sequences at the input.

OUTLINE We first describe Gaussian Mixture Regression (GMR) in Section 6.1, that utilizes Gaussian Mixture Models for regression. In Section 6.2, we extend the framework through Hidden Markov Regression (HMR), that integrates a temporal model based on Hidden Markov Models. Finally, in Section 6.3 we describe a prototype application for gestural control of physical modeling sound sythesis. Along this chapter, we will try to emphasize the power of the probabilistic representation for interaction design, and relate the issue of "regression" to that of "parameter generation".

6.1 Gaussian Mixture Regression

Gaussian Mixture Regression (GMR) takes advantages of the probabilistic modeling scheme of Gaussian Mixture Models (GMMs) for regression. The principle of the method is illustrated in Figure 6.1. It consists in learning a joint model of the motion and sound parameters. For training, we use a GMM to approximate the density over the joint space composed of the motion and sound parameters. This is illustrated by the ellipse representing the Gaussian components on the figure. For performance, we convert the joint distributions to conditional distributions. Therefore, for a given input frame of movement parameters, we can both compute the likelihood of the model on the movement only, and estimate the associated sound parameters through the distribution.

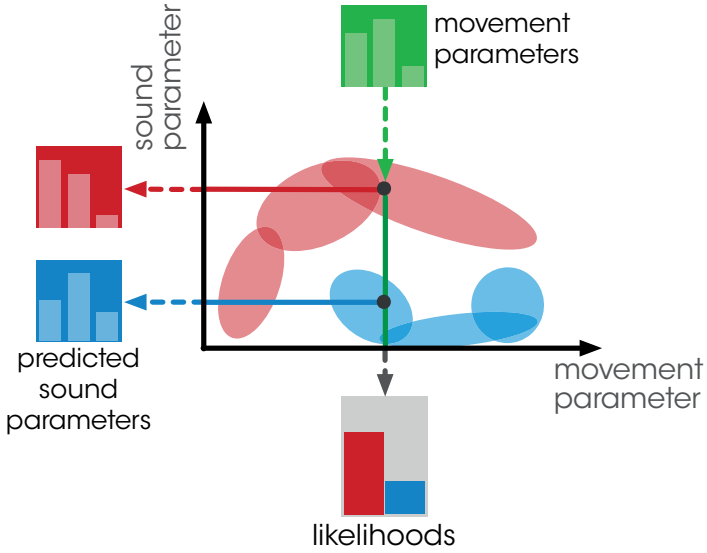


Figure 6.1: Schematic representation of Gaussian Mixture Regression. Multimodal Data is represented as a mixture of Gaussians. From new input motion, the model estimates the likelihood of the gesture and, by regression, the associated sound parameters.

Gaussian Mixture Regression (GMR) has been studied by Sung (2004) and applied to various fields of study, notably acoustic-articulatory inversion (Toda et al., 2004, 2008) and in robotics for movement trajectory generation (Calinon, 2007). It actually originates from an approach proposed by Ghahramani and Jordan (1994) aiming at learning from incomplete data via an EM approach. In this seminal work, the authors propose a method for estimating missing features in incomplete datasets by Maximum Likelihood (ML), that is, they estimate the value of missing parameters as those which maximize the joint likelihood.

We propose to use the method for sequence mapping, in which we estimate the sound parameters associated to input motion parameters as those which maximize the joint likelihood of both modalities.

6.1.1 Representation and Learning

We now detail the formulation of the model and derive the regression algorithm. We consider the input movement (m) and the output sound (s), represented by observation vectors of the form $\boldsymbol{x}^{(m)} = (x_1, \cdots, x_{D_m})$ and $\boldsymbol{x}^{(s)} = (x_1, \cdots, x_{D_s})$. The representation and training of the model is identical to a regular GMM, considering the multimodal observation vector $\boldsymbol{x}$ resulting from the concatenation of the motion and sound observation vectors:

$$\boldsymbol{x} = [\boldsymbol{x}^{(m)}, \boldsymbol{x}^{(s)}]$$

For learning, we train a GMM with a joint probability density function (pdf)

$$p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{6.1}$$

where the Gaussian parameters can be expressed as a combination of the parameters for each modality:

$$\boldsymbol{\mu}_k = \left[\boldsymbol{\mu}_k^{(m)} ; \boldsymbol{\mu}_k^{(s)}\right] \tag{6.2a}$$

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^{(mm)} & \boldsymbol{\Sigma}_k^{(ms)} \\ \boldsymbol{\Sigma}_k^{(sm)} & \boldsymbol{\Sigma}_k^{(ss)} \end{bmatrix} \tag{6.2b}$$

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.

For training, we can use a standard Expectation-Maximization (EM) algorithm, as detailed for movement models in Section 4.1.

6.1.2 Regression

CONDITIONAL DISTRIBUTION

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. The conditional density for a Gaussian distribution can be written as

$$p\left(\boldsymbol{x}^{(s)} \mid \boldsymbol{x}^{(m)}, \boldsymbol{\theta}\right) = \mathcal{N}\left(\boldsymbol{x}^{(m)} ; \hat{\boldsymbol{\mu}}^{(s)}(\boldsymbol{x}^{(m)}), \hat{\boldsymbol{\Sigma}}^{(ss)}\right) \tag{6.3}$$

where

$$\hat{\boldsymbol{\mu}}^{(s)}(\boldsymbol{x}^{(m)}) = \boldsymbol{\mu}^{(s)} + \boldsymbol{\Sigma}^{(sm)}\left(\boldsymbol{\Sigma}^{(mm)}\right)^{-1}\left(\boldsymbol{x}^{(m)} - \boldsymbol{\mu}^{(m)}\right) \tag{6.4a}$$

$$\hat{\boldsymbol{\Sigma}}^{(ss)} = \boldsymbol{\Sigma}^{(ss)} - \boldsymbol{\Sigma}^{(sm)}\left(\boldsymbol{\Sigma}^{(mm)}\right)^{-1}\boldsymbol{\Sigma}^{(ms)} \tag{6.4b}$$

We can now formulate the conditional distribution for a GMM:

$$p\left(\boldsymbol{x}^{(s)} \mid \boldsymbol{x}^{(m)}, \boldsymbol{\theta}\right) = \sum_{k=1}^{K} \beta_k(\boldsymbol{x}^{(m)}) \mathcal{N}\left(\boldsymbol{x}^{(s)} ; \hat{\boldsymbol{\mu}}_k^{(s)}(\boldsymbol{x}^{(m)}), \hat{\boldsymbol{\Sigma}}_k^{(ss)}\right) \tag{6.5}$$

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

$$\beta_k(\boldsymbol{x}^{(m)}) = \frac{w_k p\left(\boldsymbol{x}^{(m)} \mid \boldsymbol{\theta}_k\right)}{\sum_{k'} w_{k'} p\left(\boldsymbol{x}^{(m)} \mid \boldsymbol{\theta}_k\right)} \tag{6.6}$$

PARAMETER GENERATION ALGORITHM

There are several methods for generating an observation from the conditional distribution. We propose to use the Least Squares Estimate (LSE)[1] to generate the vector of sound parameters from an input vector of sound features. The estimate can be computed as the conditional expectation of $\boldsymbol{x}^{(s)}$ given $\boldsymbol{x}^{(m)}$:

$$\hat{\boldsymbol{x}}^{(s)} = E\left[\boldsymbol{x}^{(s)} \mid \boldsymbol{x}^{(m)}, \boldsymbol{\theta}\right] \tag{6.7a}$$

$$= \sum_{k=1}^{K} \beta_k(\boldsymbol{x}^{(m)}) \hat{\boldsymbol{\mu}}_k^{(s)}(\boldsymbol{x}^{(m)}) \tag{6.7b}$$

$$\bar{\boldsymbol{\Sigma}}^{(s)} = \sum_{k=1}^{K} \beta_k(\boldsymbol{x}^{(m)})^2 \hat{\boldsymbol{\Sigma}}_k^{(s)}(\boldsymbol{x}^{(m)}) \tag{6.7c}$$

The Least Squares Estimate (LSE) takes into account the contribution of all the components in the mixture model. Interestingly, this estimator can be seen as a convex sum of linear regressions where the weights vary dynamically over the input space according to the responsibility of each component for the observed data.

Several other methods have been proposed in the literature (Ghahramani and Jordan, 1994; Toda et al., 2008). Alternatively, single component Least Squares Estimate (LSE) considers the expectation of the likeliest component only. Stochastic Sampling (STOCH) is another alternative that consists in randomly sampling the conditional distribution. Finally, Maximum Likelihood (ML) estimation (Toda et al., 2008) determines the value of the output observation that maximizes the conditional pdf.

STOCH and single component LSE can produce very noisy results, while LSE smooths between the components of the mixture. Similarly, the ML estimate tends to introduce abrupt changes — except when using delta features, which requires to solve for an entire sequence and is consequently incompatible with online inference, — and is computationally more intensive that the LSE. Therefore, for the purpose of continuous sonic interaction, LSE is the most relevant estimator for its smoothness and low computational cost.

6.1.3 Number of Components

The implementation derives from the GMM implementation presented in Section 4.1. It integrates the same parameters of number of Gaussians, regularization, and Likelihood Window.

1 also called Minimum Mean Square Error (MMSE) (Toda et al., 2008)

For movement models, the number of Gaussian components specifies the complexity of the model. It allows to define non-linear boundaries between classes as the Gaussian components fit the regions of support of the input space.

In the case of multimodal modeling, the number of components has the same implications for both the input and output modalities. Most importantly, it conditions the shape and complexity of the relationship between the motion and sound parameters. While using a single Gaussian component results in a linear relationship, the non-linearity of the mapping increases with the number of components. The notion of complexity in this method differs from functional regression approaches in which complexity is often conditioned by the order of the function. Here, it is defined by the shape of the distribution over multimodal data, introducing non-linear behaviors only in the relevant regions of support.

An example of the influence of the number of Gaussian components is illustrated in Figure 6.2 on synthetic data. We learn 3 GMR from a single example that puts in relationship two modalities. In the example, the input modality is only a time vector and the output modality presents a complex response that spans from a slowly evolving bell shape to rapid oscillations. The figure depicts the training example along with the 95% confidence ellipses of each Gaussian component. The resynthesis of the output modality from the same time vector using GMR is plotted in red on the figure.

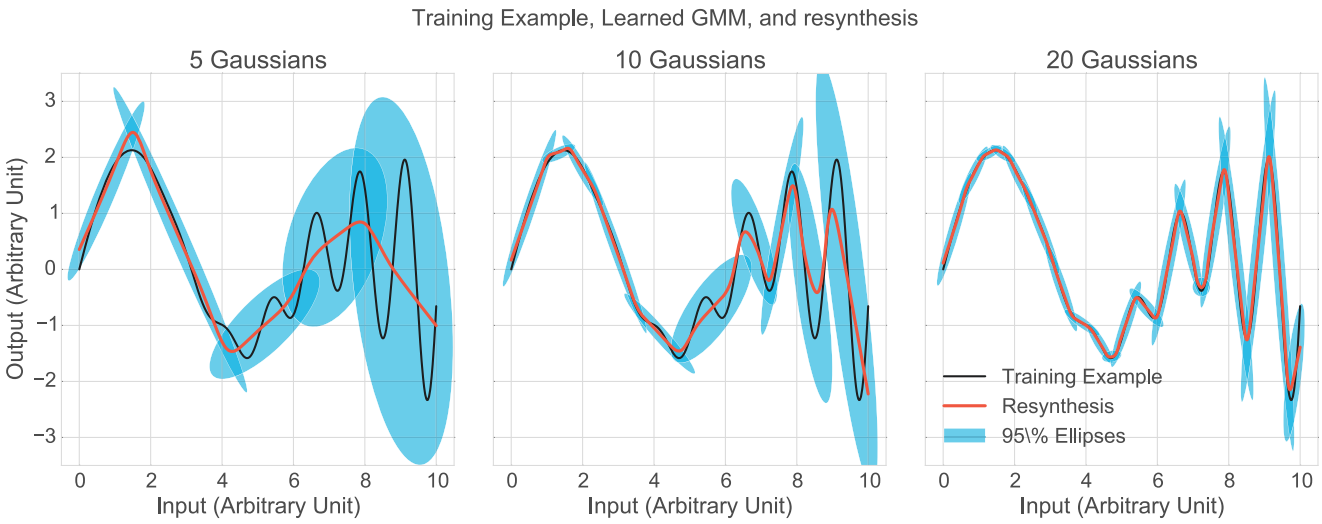


Figure 6.2: Example of Multimodal GMM and Gaussian Mixture Regression on artificial data. 5, 10 and 20 components are respectively used for comparison. The models were trained with regularization $\lfloor\sigma\rfloor = 5e-4$

Learning a regression with only 5 components (left plot) results in a very rough approximation of the example gesture. In this case, the oscillations are considered to be noise as the model complexity is too low to account for these variations. Increasing the number of Gaussians results in a better encoding of the oscillatory behavior (middle plot), although 10 components are not sufficient to model these profiles accurately and result in under-

shooting of the extrema. Using 20 mixtures (right plot) provides a correct fit to the training example, and enables to resynthesize accurately the output modality.

It is essential to note that while the second part of the gestures presents higher frequency content, it is represented with less Gaussians than the beginning of the gesture. This example highlights two properties of GMR. First, we observe that the overlap between the Gaussian components allows to interpolate the linear portions to generate non-linear behavior. Second, more components are needed to model non-linear behaviors such as the slowly evolving shape at the beginning of the example.

6.1.4 Regularization

We introduced a regularization strategy for GMMs in Section 4.1.4, that adds a prior to the variances of each Gaussian component. For movement models, this variance prior allows to regularize the recognition boundaries by increasing the overlap between components within a model, and possibly between classes. For GMR, regularization is crucial to determine the smoothness of the regression function, as it impacts the overlap between adjacent Gaussians.

Using the example introduced previously (Figure 6.2, we now illustrate the influence of regularization on synthetic data. We trained 3 GMMs with 20 Gaussians components using varying levels of regularization; the training example, learned model, and resynthesis are plotted in Figure 6.3.

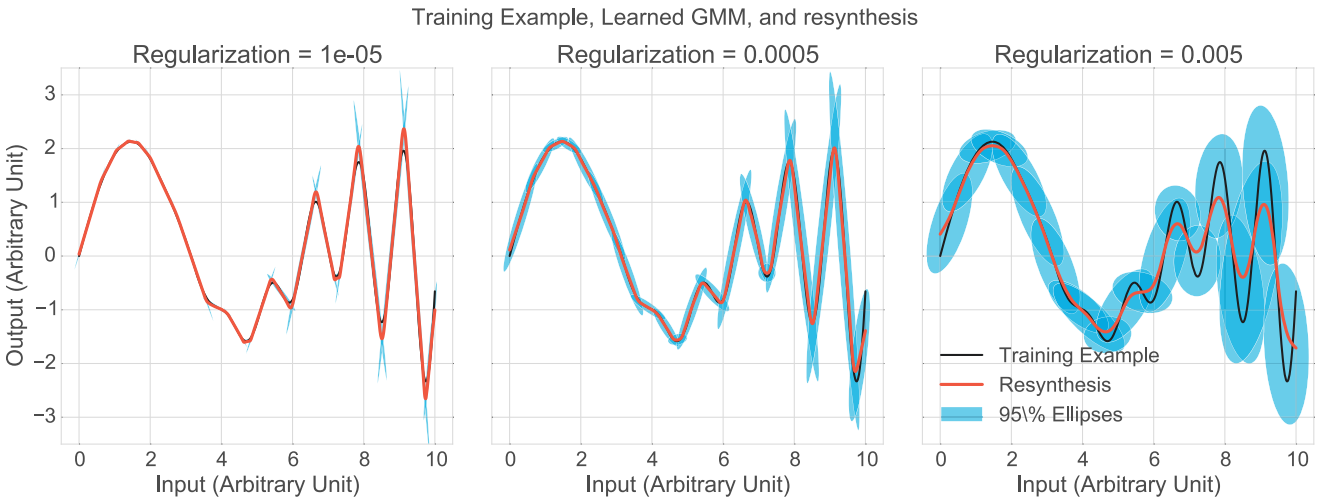


Figure 6.3: Example of Multimodal GMM and Gaussian Mixture Regression on artificial data. Various levels of regularization are used for comparison. The models were trained with 20 Gaussian Components.

Using very small regularization ($\lfloor\sigma\rfloor = 1e^{-5}$; Figure 6.3, left) tends to approximate the training data by piecewise linear regression. Indeed, the low values of the variance exclude almost all overlap between the components; as a result, the resynthesis approaches the training example by concatenating linear segments, which result in overshooting the extrema of the end

of the gesture. At the other end of the spectrum, a relative regularization of $\lfloor\sigma\rfloor = 5e^{-3}$ (Figure 6.3, right) gives a very smooth resynthesis of the output modality. As the overlap between components increases, the regression function tends to undershoot the targets. An intermediate value of $\lfloor\sigma\rfloor = 5e^{-4}$ (Figure 6.3, middle) represents a good compromise that allows to resynthesis the output accurately while guaranteeing a fairly smooth interpolation between Gaussians.

Therefore, the impact of regularization goes beyond the single issue of avoiding overfitting, but also has a strong impact on the smoothness of the estimated regression function. Our implementation fits the constraints of Interactive Machine Learning, and features regularization as an essential parameters allowing users to adjust the properties of mapping models trained on small datasets.

6.1.5 Discussion

Gaussian Mixture Regression (GMR) provides a flexible framework for motion-sound mapping. The method is based on sound parameter generation algorithms that draw upon a density distribution over the sound parameters that is conditioned on the motion parameters.

The power of the model resides in its semiparametric approach to regression. Indeed, instead of estimating the parameters of an arbitrary regression function, GMR draws upon the estimation of a density function over multimodal training data. Regression can be performed by expressing the output modality, sound, in a conditional probability density function where movement is observed. The complexity of the regression function is therefore determined by the number of components in the model. The most interesting aspect of the method is that GMMs can approximate arbitrary densities by focusing on the regions of support of the input space. The support and level of detail of the mapping function can therefore be authored transparently by the users,.

6.2 Hidden Markov Regression

In this section we introduce a regression scheme that combines the Gaussian Mixture Regression method described in Section 6.1 with HMM-based sequence modeling.

Along this work, we call this approach Hidden Markov Regression (HMR), in echo to the equivalent designation for GMMs. In previous work, we referred to Hidden Markov Regression (HMR) as Multimodal Hidden Markov Model (MHMM). In other works, authors refer to similar methods as *multimodal HMM* (Hofer, 2009), *HMM inversion* (Choi et al., 2001), *cross-modal HMM* (Fu et al., 2005), to name a few.

PRINCIPLE Figure 6.4 illustrates the process of mapping with HMR. The method is similar to GMR, but with the integration of a sequence model. We start by learning a HMM on joint multimodal data; i.e. , 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 estimate the likelihood of each model, and to generate the associated sound parameters from the conditional distribution.

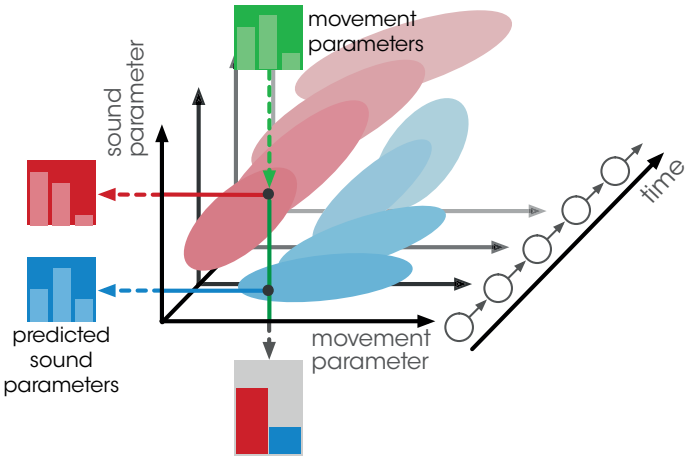


Figure 6.4: Schematic representation of Hidden Markov Regression. Multimodal Data is represented as a Hidden Markov Model. From new input motion, the model estimates the likelihood of the gesture and, by regression, the associated sound parameters, given the common temporal structure of motion and sound.

6.2.1 Representation and Learning

As for GMR, the representation utilizes the standard HMM representation with multimodal features. We learn a HMM with the EM algorithm on a set of training examples whose feature vector concatenate movement features and sound parameters. The joint model can be written using Equation 4.6 using the joint feature vectors $\boldsymbol{x} = [\boldsymbol{x}^{(m)}, \boldsymbol{x}^{(s)}]$. The observation model therefore becomes

$$p(\boldsymbol{x} \mid z_t = j) = \mathcal{N}\left(\boldsymbol{x}; [\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\mu}_j^{(s)}], \begin{bmatrix} \boldsymbol{\Sigma}_k^{(mm)} & \boldsymbol{\Sigma}_k^{(ms)} \\ \boldsymbol{\Sigma}_k^{(sm)} & \boldsymbol{\Sigma}_k^{(ss)} \end{bmatrix}\right) \tag{6.8}$$

With such a formulation, we make the assumption that both movement and sound are generated by the same underlying Markov process. We estimate the parameters of this process with a joint representation motion and sound parameters.

Other methods have been proposed for training such cross-modal models. Brand (1999) proposed HMM remapping for speech-driven character animation, which consists in training a HMM on a single modality, then

remapping the observation distributions to the other modality. However, this method proved less efficient than joint HMMs for audio-visual mapping (Fu et al., 2005). Recently, Wu and Wang (2006) proposed a minimum generation error training procedure for HMM-based speech synthesis. However, the method relies on a MAP estimation of the state sequence, that is incompatible with online inference. As a first step, the EM algorithm appears the best choice for training the model representing motion and sound jointly.

6.2.2 Inference and Regression

PARAMETER GENERATION ALGORITHMS

Several methods have been proposed for sequence mapping with Hidden Markov Models. Initial techniques derive from speech synthesis methods that rely on a fixed state sequence to generate the output observations (Tokuda et al., 2000).

For audio-visual sequence mapping, Chen (2001) proposed to estimate the optimal state sequence from the input modality using Maximum A Posteriori (MAP). In the synthesis step, they perform GMR using the observation model of the current state estimated with the Viterbi algorithm. The method has several shortcomings. First, it requires the entire observation sequence to perform Viterbi decoding. Second, we argue that using a fixed state sequence leads to poor synthesis results, as decoding errors propagate to the synthesis step. Moreover, when using a single-Gaussian observation model, the method amounts to piecewise linear regression without guarantee of the continuity between the linear segments.

Alternatively, Choi et al. (2001) derived an iterative estimation of the output sequence based on the Expectation-Maximization (EM) algorithm, that maximizes the joint probability of both input and output modalities. Their method iteratively estimates the optimal sequence through expectation (estimation of posterior state probabilities using the multimodal sequence) and maximization (estimation of the output sequence from posteriors). This technique, called *HMM Inversion*, was found smoother and more accurate than the MAP-based method on audio-visual mapping (Fu et al., 2005).

Nonetheless, neither approach is adapted to the context of continuous interaction. Both techniques require the entire input sequence to be available to generate the sequence of output parameters.

PROPOSED METHOD

We propose a method based on a filtered estimation of state probabilities. The estimate is the causal Maximum Likelihood estimate that can be computed in real-time, generating the sound parameters as soon as the frame of motion parameters is available.

We start by expressing the joint model as a conditional model where the density over the output modality (s) is expressed conditionally to the input feature modality (m). The conditional density over the sequence of sound parameters can be expressed as

$$p(\boldsymbol{x}_{1:T}^{(s)}|\boldsymbol{x}_{1:T}^{(m)},\lambda) = \prod_{t=1}^{T}\left[\sum_{i=1}^{N} \overbrace{p(\boldsymbol{x}_t^{(s)} \mid z_t = i, \boldsymbol{x}_t^{(m)},\lambda)}^{\text{Conditional observation model}} \underbrace{p(z_t = i \mid \boldsymbol{x}_{1:T}^{(m)},\lambda)}_{\text{Posterior state likelihood}}\right] \tag{6.9}$$

For online estimation, we can thus estimate the distribution over output features recursively:

$$p(\boldsymbol{x}_t^{(s)}|\boldsymbol{x}_{1:t}^{(m)},\lambda) = \sum_{i=1}^{N} p(\boldsymbol{x}_t^{(s)} \mid z_t = i, \boldsymbol{x}_t^{(m)},\lambda)\, p(z_t = i \mid \boldsymbol{x}_{1:t}^{(m)},\lambda) \tag{6.10}$$

which can be simplified as[2]

$$p(\boldsymbol{x}_t^{(s)}|\boldsymbol{x}_{1:t}^{(m)},\lambda) = \sum_{i=1}^{N} \mathcal{N}\left(\boldsymbol{x}^{(s)}; \hat{\boldsymbol{\mu}}_i^{(s)}(\boldsymbol{x}_t^{(m)}), \hat{\boldsymbol{\Sigma}}_i^{(ss)}\right) \alpha_t^{(m)}(i) \tag{6.11}$$

where $\alpha_t^{(m)}(i)$ is the forward variable estimated on the movement features only, and where $\hat{\boldsymbol{\mu}}_i^{(s)}(\boldsymbol{x}_t^{(m)})$ and $\hat{\boldsymbol{\Sigma}}_i^{(ss)}$ are the mean and covariance of the conditional Gaussian distribution, as defined in Equation 6.4.

We utilize the filtered state marginals as weights for Gaussian Mixture regression. The prediction is therefore averaging over all possible state sequences, rather than choosing the MAP estimate, which yields a smoother synthesis. The method is similar to that of Calinon et al. (2010) for movement generation in robotics.

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

$$\hat{\boldsymbol{x}}_t^{(s)} = E\left[\boldsymbol{x}_t^{(s)} \mid \boldsymbol{x}_{1:t}^{(m)}, \boldsymbol{\theta}\right] \tag{6.12a}$$

$$= \sum_{i=1}^{N} \alpha_t^{(m)}(i)\hat{\boldsymbol{\mu}}_k^{(s)}(\boldsymbol{x}_t^{(m)}) \tag{6.12b}$$

$$\bar{\boldsymbol{\Sigma}}_t^{(s)} = \sum_{i=1}^{N} \alpha_t^{(m)}(i)^2 \hat{\boldsymbol{\Sigma}}_k^{(s)}(\boldsymbol{x}_t^{(m)}) \tag{6.12c}$$

6.2.3 Example

We propose to illustrate the process of Hidden Markov Regression with a synthetic example. We consider abstract motion and sound parameters,

2 For simplicity, we considered a single Gaussian per observation distribution. Extending the method to an observation model defined as a Gaussian mixture is straightforward using the GMR formalism.

and we propose to compare the ability of GMR and HMR to resynthesize the trajectory of the sound parameter from the motion parameter sequence. Figure 6.5 illustrates the training example (black line) composed of the sequence of motion parameters and the sequence of sound parameters. We train a GMR and a HMR on this sequence, with 20 mixture components, and 20 states, respectively.

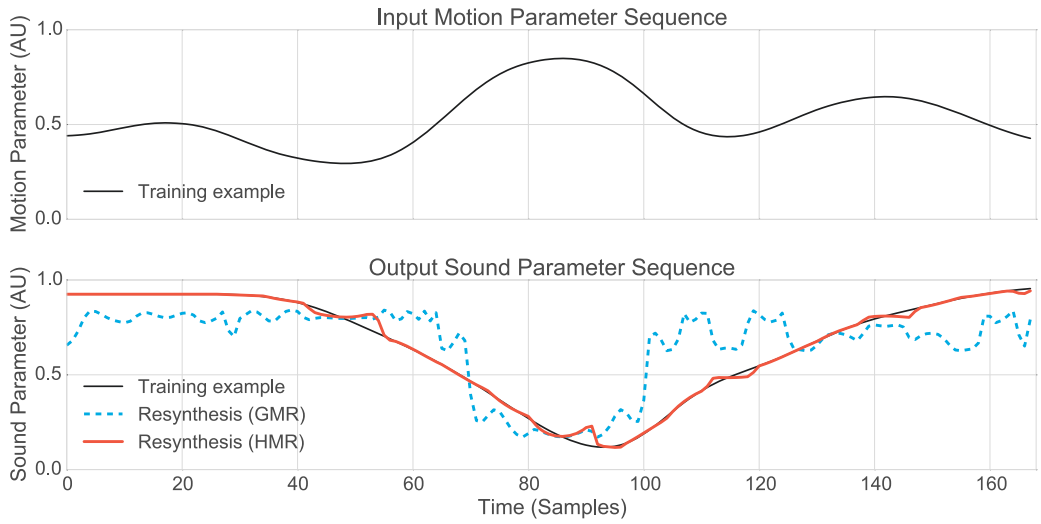


Figure 6.5: Example of regression with GMR and HMR. The GMR and HMR models were trained with 20 Gaussians and 20 states, respectively.

In a second step, we stream the same motion parameter sequence to each model to attempt to resynthesize the associated sequence of sound parameters. The resyntheses using GMR and HMR are presented by the dashed blue and red trajectories in Figure 6.5. We observe that in this case, GMR poorly reconstructs the original sound parameter trajectory.

This example highlights a limitation of Gaussian Mixture Regression, that relates to the complexity of the mapping. This example presents a very complex mapping, that does not have a one-to-one correspondence between values of the motion parameters, and values in the sound parameter space — in other word, there is no 'functional' relationship between the input motion and the sound output.

A spatial representation of the relationship between the motion and sound parameters is represented in Figure 6.6, along with the resyntheses computed with each model. We observe that in the training example, several values of the sound parameter are associated to different values of the motion parameters, depending of the context in the sequence.

GMR sums the contribution of each Gaussian depending on its contribution to the likelihood over the input space. Therefore, in this case GMR fails at reconstructing the trajectory and results in intermediate values of the sound parameters.

On the contrary, HMR exploits the sequence model of HMMs for modeling both the input and output processes. The weights of the Gaussians for

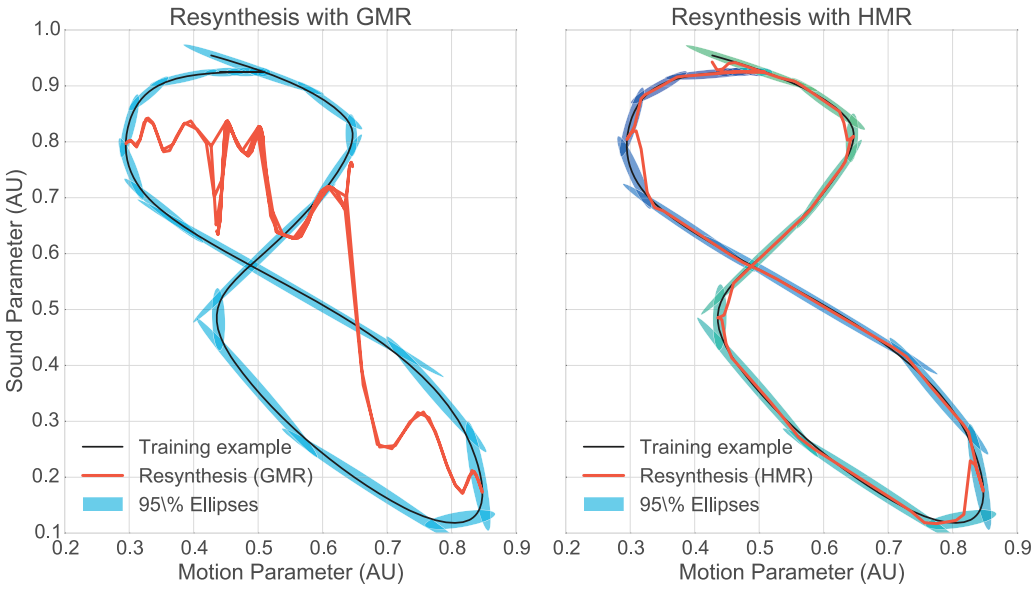


Figure 6.6: Spatial representation of the regression with GMR and HMR. The GMR and HMR models were trained with 20 Gaussians and 20 states, respectively. Shaded ellipses represent the 95% confidence interval of the Gaussian components. In the case of HMR, the ellipses are colored from blue to green according to their position in the left-right hidden Markov chain.

regression are defined by the state probabilities that encode the temporal context of the sequence.

This simple example presents an extreme case of complex mapping, and does not aim to represent a meaningful mapping between motion and sound. However, it illustrates that HMR has a powerful representation of the context. HMR exploits the context of the sequence to address the possible ambiguities of the demonstrated relationship between motion and sound, and guarantees a better consistency of the generated sound parameters.

6.2.4 Number of States and Regularization

Similarly to our implementation of HMMs for movement modeling, the user has access to the number of states and regularization. These parameters help defining the shape and properties of the relationship between motion and sound. As for HMMs, the number of states in HMR specifies the complexity of the sequence model. For regression, this impacts the accuracy with which the trajectories are sampled by the temporal model, which has an effect on the properties of the mapping.

Figure 6.7 illustrates the influence of the number of states in the simple example introduced in the previous section. We trained two HMR with respectively 10 and 20 states, and we use the motion feature sequence used for training to resynthesize the associated sequence of sound parameters.

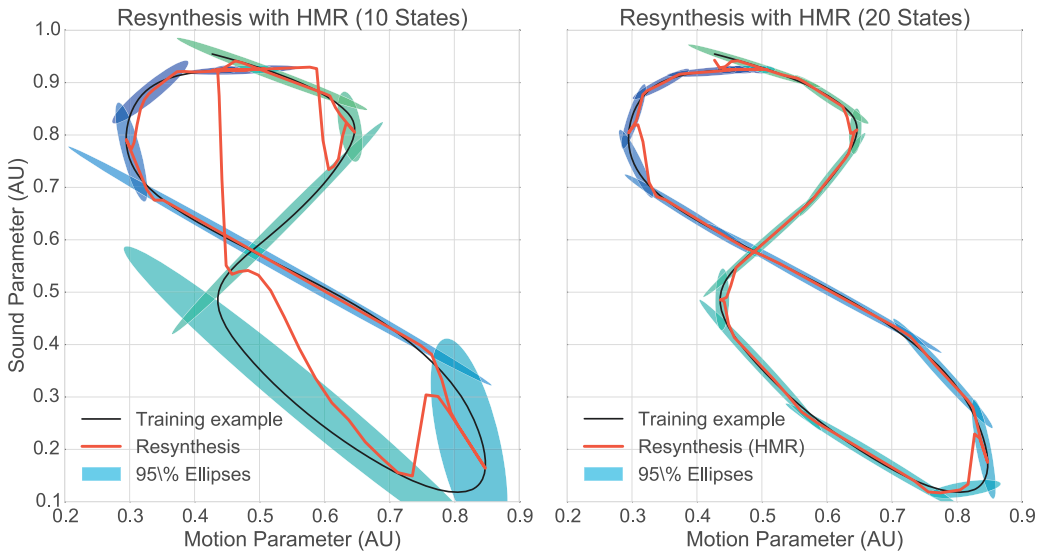


Figure 6.7: Influence of the Number of Hidden States on HMR

We observe that the number of states is critical to context modeling. Using 10 states (Figure 6.7, left) does not allow for synthesizing the trajectory with accuracy. There is a large overlap between the state distributions, in particular at the end of the gesture. During regression, the state probabilities propagate along the Markov chain, which implies a great overlap between the components at the end of the gesture. Increasing the number of states to 20 (Figure 6.7, right) allows to better reproduce the demonstrated trajectory, as the states become more discriminative on the input motion. However, a too large number of hidden states might limit the possibility of generalizing the demonstrated relationship to novel regions of the space of motion parameters.

As for GMR, regularization impacts on the overlap between the Gaussian components, which can help increasing the smoothness of the generated sound trajectories. However, a large regularization can be detrimental to the sequence model and decrease the accuracy of the learned relationship.

6.2.5 Hierarchical Hidden Markov Regression

We proposed in Section 4.5 an extension of HMMs implementing a segmental representation of gestures. The Hierarchical Hidden Markov Model (HHMM) extends simple HMMs by embedding motion segments in a higher level transition structure.

Extending HMR to the hierarchical model is straightforward. Once again, we can take advantage of the GMR formalism. The sound parameter generation can be performed similarly to a single HMR, replacing the state probabilities $\alpha_t(i)^{(m)}$ by the filtered marginals $\alpha_t(i,c)^{(m)}$ estimated with the forward algorithm of the HHMM, where c is the label of the current segment.

The hierarchical structure allows to improve the continuous recognition of particular gestures. It integrates a representation of high level depen-

dencies between motion segments that account for long-term contextual information.

WINDOWING STATE POSTERIORS

The hierarchical structure of the HHMM introduces a possible problem of consistency for sound parameter generation. As discussed in Section 4.6.1, the HHMM introduces a mechanism of exit states that activates the initial probabilities of the segments when the end of the current segment is reached.

This implies that the state probabilities can possibly propagate from the last states of a motion segment to the beginning of the same segment. This situation is illustrated in the bottom plot of Figure 6.8, that represents the posterior state probabilities. In such a situation, the parameter generation algorithm will generate the sound parameters by mixing the sound parameters at the beginning and at the end of the segment, which might result in inconsistencies.

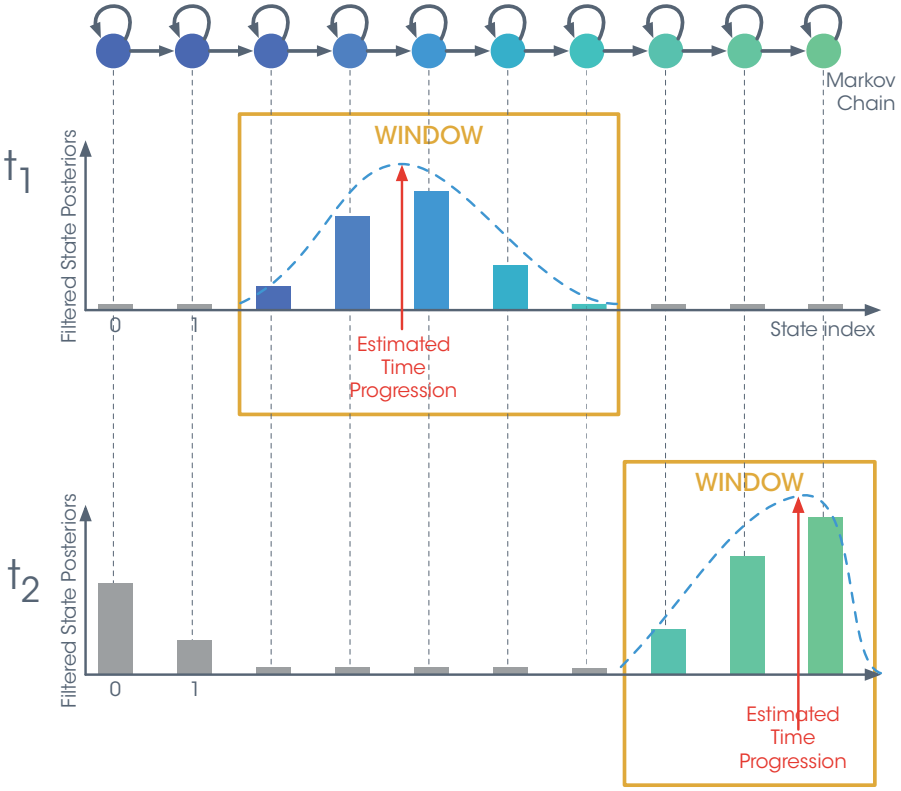


Figure 6.8: Posterior windowing strategy for guaranteeing sound parameter.

We propose to address this problem using a windowing of the state posteriors. For regression, we only consider a subset of the states in a window centered around the likeliest state, and bounded by the first and last state. In practice, the window size is half the number of hidden states, and we use for regression only the states i which match the following condition:

$$\max[0;\ i_{max} - N/2] \quad \leq \quad i \quad \leq \quad \min[i_{max} - N/2;\ N] \tag{6.13}$$

where i_{max} is the index of the likeliest state and N is the total number of states.

The regression is then performed using the LSE on this reduced distribution of Gaussians. The process is illustrated in Figure 6.8 for two possible distributions of the state probabilities. At t_1, the performance is at the middle of the gesture. In this case we consider the window centered on the likeliest state. At t_2, the performance is reaching the end of the segment, and the probabilities start to propagate at the beginning of the segment. In this case we consider a window that is bounded at the end of the segment, which guarantees that the estimate of the sound parameters does not interpolate between the beginning and the end of the segment.

6.3 Illustrative Example: Gesture-based Control of Physical Modeling Sound Synthesis

In this section, we present a use-case in gestural control of sound synthesis that uses Hidden Markov Regression (HMR)[3] for learning the relationship between gestures, captured with accelerometers, and sequences of sound parameters of a physical model. This example aims to illustrate how the proposed method can be used in a practical application in sound control. Our application uses the Max/MuBu implementation of the XMM library for Hidden Markov Regression (see Appendix A.2).

This section is an adaptation of a previous publication: "Gesture-based control of physical modeling sound synthesis: a Mapping-by-Demonstration Approach", presented at the ACM International Conference on Multimedia in 2013 (Françoise et al., 2013a). The article is a demonstration proposal supporting a short paper introducing Hidden Markov Regression for gesture–sound mapping (Françoise et al., 2013b). For clarity, and to avoid interactions with the theoretical aspects of the modeling framework, we chose to adapt the article rather than reporting the full publication.

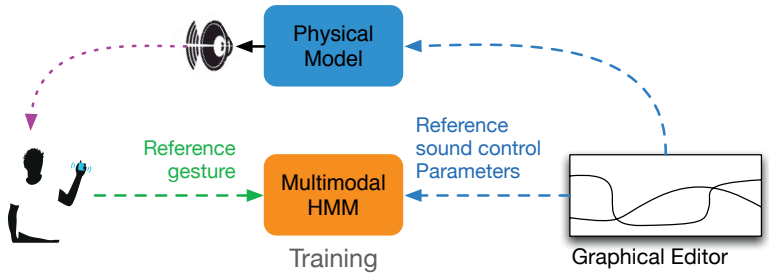
6.3.1 Motion Capture and Sound Synthesis

The applications maps between movements captured with *Modular Musical Objects* (MO) for motion capture (Rasamimanana et al., 2011) with physical modeling sound synthesis. Our system uses *Modalys*[4] (Causse et al., 2011), a software dedicated to modal synthesis, i.e. that simulates the acoustic response of vibrating structures under an external excitation. It allows to build virtual instruments by combining *modal elements* — e.g. plates, strings, membranes — with various types of connections and exciters — e.g. bows, hammers, etc. Each model is governed by a set of physical parameters — e.g. speed, position and pressure of a bow. Specific sounds and playing modes can be created by designing time profiles combining these parameters.

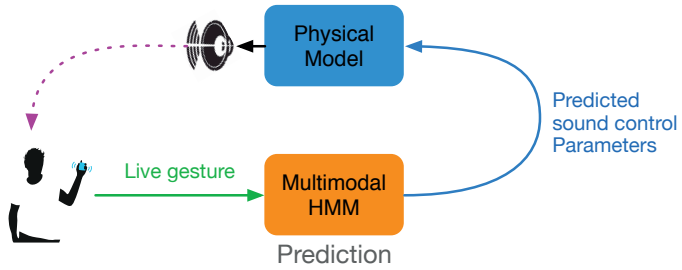
3 Note that in the original article, Hidden Markov Regression was called Multimodal HMMs.
4 http://forumnet.ircam.fr/product/modalys/

6.3.2 Interaction Flow

The workflow of the application is an interaction loop integrating a *training* phase and a *performance* phase. It is illustrated in figure 6.9, and a screenshot of the software is depicted in figure 6.10. In the *training* phase,



(a) *Training*: sounds are designed using a graphical editor, and reference gestures can be recorded while listening.



(b) *Performance*: the model is used to predict the sound parameters associated with a live gesture.

Figure 6.9: Application workflow.

the user can draw time profiles of control parameters of the physical models to design particular sounds. Each of these *segments* can be visualized, modified, and played using a graphical editor (top left of figure 6.10). Then, the user can perform one or several demonstrations of the gesture he intents to associate with the sound example (figure 6.9a). Gesture and sound are recorded to a multimodal data container for storage, visualization and editing (bottom left of figure 6.10). Optionally, segments can be manually altered using the user interface. The multimodal HMM representing gesture–sound sequences can then be trained using several examples. During the *performance* phase, the user can gesturally control the sound synthesis. The system allows for the exploration of all the parameter variations that are defined by the training examples. Sound parameters are predicted in real-time to provide the user with instantaneous audio feedback (figure 6.9b). If needed, the user can switch back to *training* and adjust the training set or model parameters.

A video that demonstrates the use of the system is available online.[5]

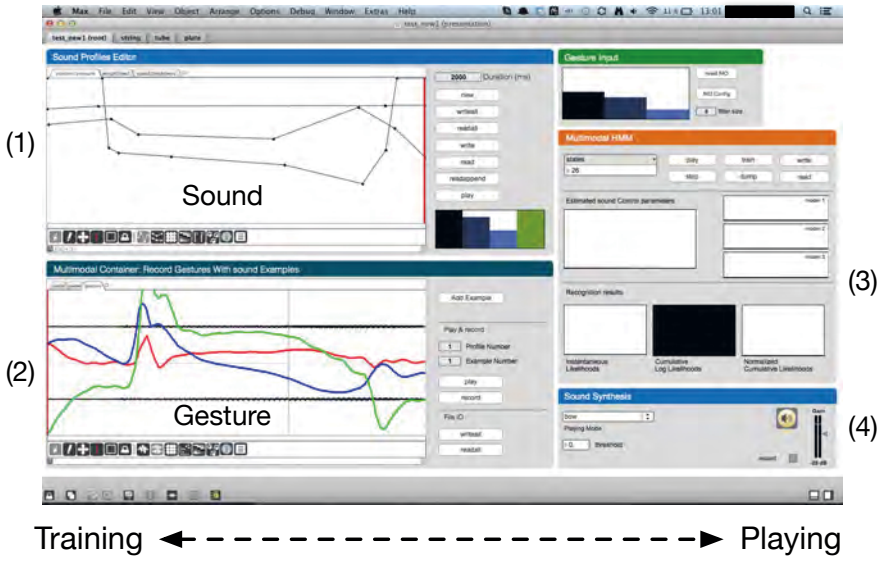


Figure 6.10: Screenshot of the system. **(1)** Graphical Editor. **(2)** Multimodal data container. **(3)** Multimodal HMM: control panel and results visualization. **(4)** Sound synthesis.

6.4 Discussion

We proposed two approaches to motion-sound mapping using multimodal probabilistic models. Stochastic approaches to feature mapping draw upon the estimation of density distribution over a joint space composed of motion and sound parameters. The translation of the joint model in a conditional model, that expresses the distribution over sound parameters conditionally to the input motion features, can be used to perform the mapping. The method therefore brings a probabilistic formulation of the mapping.

This probabilistic approach gives an original perspective on regression, as the interaction model is based on density estimation rather than functional approximations. The mapping model is therefore grounded in the regions of support of the motion and sound parameter spaces. The probabilistic approach allows to model uncertainty on both sound and motion, consistently encoding uncertainty in the input-output relationships. The Gaussian formalism makes the framework parametric and extensible.

In this section, we discuss two issues related to such a probabilistic formulation of the mapping. First, we illustrate how the models can generate not only the sound parameters, but also their associated covariances that define the uncertainty of the synthesized parameter trajectories. Second, we discuss possible strategies for combining mappings when several classes are available.

5 http://vimeo.com/julesfrancoise/mm13

6.4.1 Estimating Output Covariances

The advantage of Gaussian models such as GMMs and HMMs is their representation of uncertainty in the covariance of the Gaussian distributions. Both GMR and HMR can predict the variance associated with the estimation sound parameters during parameter generation. Although we do not fully exploit such variances in this thesis, we believe that they represent an additional parameters usable for sound control, that represent the confidence over the generated sound parameters.

EXAMPLE Figure 6.11 illustrates the estimation of the variances over the generated sound parameters in the case of GMR. The demonstration is reported in the left plot, where a single training example is approximated by a multimodal mixture of Gaussians. The right plot presents the resynthesis of the sound feature sequence from the same motion feature sequence. The right plot depicts the estimated standard deviation for each generated parameter as a function of the input value. It expresses the confidence in the generated sound parameter vector, and could be used as an additional parameter for controlling sound synthesis.

We describe an example of application of the output covariance estimation in Chapter 7. We consider how the variances can be used to investigate performers' consistency across trials of a known movement sequence.

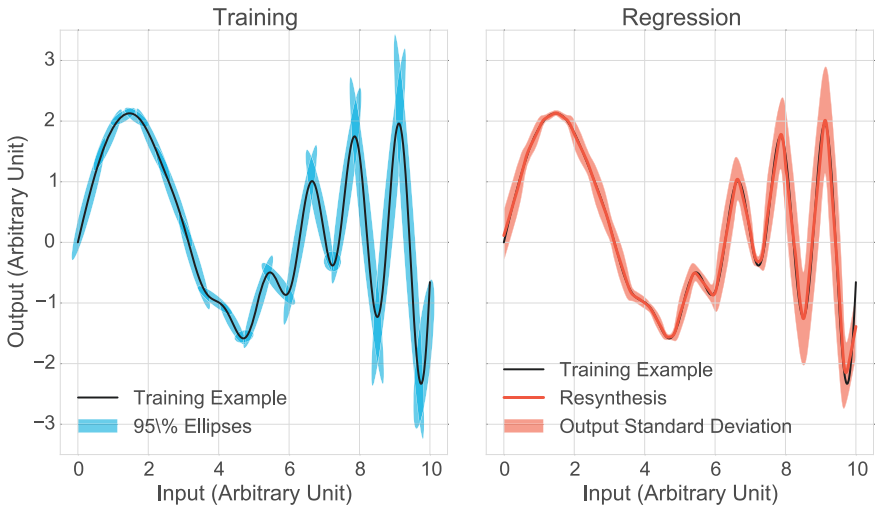


Figure 6.11: Output variance estimation with GMR

6.4.2 Strategies for Regression with Multiple Classes

Often, one might want to implement several classes of gestures in relation to sounds. These classes might relate to particular gestures. Our implementation of both GMR and HMR provides a flexible interface for handling multiple classes in regression problems. We propose three strategies for sonic interaction design that compromise between classification and con-

tinuous recognition. The three strategies — Parallel Processing, Classification, and Mixture of Models — are summarized in Figure 6.12.

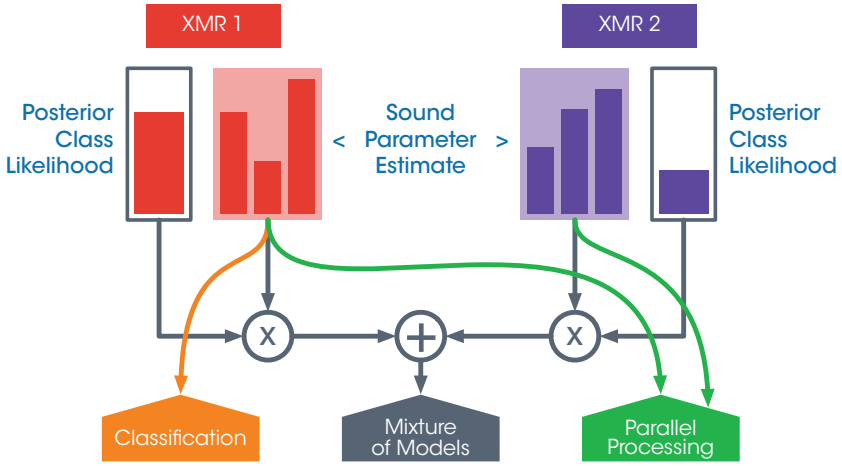


Figure 6.12: Summary of the three strategies for Regression with multiple Classes. XMR represents a regression model, either GMR or HMR.

PARALLEL PROCESSING The first strategy consists in considering the estimates of all the models simultaneously. For examples, each class can be used to control a synthesis engine independently. Using all the estimates in parallel therefore allows to superimpose mapping strategies. This design strategy can be complemented by a control based on the likelihood of each class, e.g. using the class posteriors to continuously fade between the synthesizers.

CLASSIFICATION Classification relates to the set of applications where strict gesture recognition is desired: a single gesture must be recognized to activate its associated mapping. In this case the solution is trivial: posterior classes likelihoods are used to perform recognition, and the mapping associated with the likeliest model only is used to estimate the output features. Formally, this process can be described by

$$\hat{\boldsymbol{x}}^{(s)} = \sum_{c} \delta\left(c, \underset{c'}{\operatorname{argmax}}\, p(c'|\boldsymbol{x})\right) \hat{\boldsymbol{x}}^{(s)}_{|c} \tag{6.14}$$

where $\hat{\boldsymbol{x}}^{(s)}_{|c}$ is the Least Squares Estimate for class c.

MIXTURE OF MODELS Alternatively, it is possible to leverage the contribution of each class by combining the estimates of each class. This leads to combine the predictions using a weighting by their posterior class likelihood. Formally, we are creating a mixture of GMMs experts, that can be expressed as a GMM that aggregates all the components of each class with a weighting based by the posterior probabilities. In practice, the LSE can be expressed from the estimate of each class as

$$\hat{\boldsymbol{x}}^{(s)} = \sum_{c} p(c|\boldsymbol{x}) \hat{\boldsymbol{x}}^{(s)}_{|c} \tag{6.15}$$

This estimate allows to define a unique regression function by agglomerating a set of classes. This method allows to smoothly interpolate between the results of each model.

6.5 Summary and Contributions

This section introduces generative probabilistic models for motion-sound mapping. We present two methods for learning the relationships between sequences of motion and sound parameters that draw upon the estimation of a multimodal density distribution. Gaussian Mixture Regression (GMR) uses a Gaussian Mixture Model trained on joint recordings of motion and sound parameter sequences to build a regression model that expresses the distribution over sound parameters conditionally to the input motion features. Hidden Markov Regression (HMR) combines GMR with the sequence model of Hidden Markov Models. We propose an online estimation algorithm for HMR based on the Least Squares Estimate that allows to perform the mapping in real-time. HMR brings a contextual representation of the relationships between motion and sound that can improve the consistency of the sound parameter generation with respect to the demonstrated mapping.

7

Movement and Sound Analysis using Hidden Markov Regression

We proposed in Chapter 5 a methodology for online analysis of movement performances based on continuous recognition and alignment. We now consider the other facet of probabilistic models: generation. We propose to use the probabilistic models for parameter generation presented in Chapter 6 for analyzing movement performance and vocalized movements.

The approach based on probabilistic movement models, described in Chapter 5 ,provides tools for comparing individual performances. In this chapter, we investigate how the internal structure of models learned from several examples can give insights into the consistency or variability of a performer across trials. We propose a method synthesizing trajectories of motion parameters, that we apply to movement analysis, both within performer and between different persons. We then propose a methodology for analyzing participants vocalizations performed while moving.

7.1 Methodology

The internal parameters of Hidden Markov Models (HMMs), such as the transition matrix or the mean and covariance of the Gaussian observation models, have a transparent interpretation. Schematically, HMMs embed a continuous trajectory of movement parameters in a set of hidden states that describe its temporal evolution. For movement analysis, we could directly investigate the mean and covariance values of each state of the model. However, we prefer using the models for generation as a way to visualize their internal representation. We believe that movement synthesis is easier to interpret, as relates more easily to the observed sequences of motion parameters.

OVERVIEW OF THE APPROACH

The proposed method consists in learning a Hidden Markov Regression (HMR) Model between a time vector and sequences of motion features. Learning such a mapping between time and motion parameters allows to resynthesize trajectories of motion features from new input time vectors. This method was proposed for movement generation in robotics by Calinon that used GMR Calinon et al. (2007), HMMs Calinon et al. (2010), and HSMMs Calinon et al. (2011). Here, our focus is not as much on movement generation for animation or robotics as it aims to provide an analytics tool for movement study.

7.1.1 Dataset

In this section, we consider the dataset of Tai Chi movements we already introduced in Section 5.1. As a reminder, we study various performances of the same known sequence of Tai Chi movements by two performers with varying expertise. Both Performers **T** (Teacher) and **S** (Student) recorded 10 trials of a sequence of approximately 13 gestures, and about 45 seconds long. Each sequence was manually annotated in a set of 12 or 13 motion segments. For more details, refer to Section 5.1.

7.1.2 Time-based Hierarchical HMR

We train a Hierarchical HMR model from a set of segments extracted from all N trials of a given participant. The motion is represented by the 3D acceleration captured with the sensor placed on the Sword. Each trial n is segmented in 12 or 13 temporally ordered classes using the manual annotation to constitute a set of motion segments

$$\boldsymbol{x}^{(n,i)} = \boldsymbol{x}_{t_i:t_{i+1}} \tag{7.1}$$

where t_i is the start index of the ith segment. We then associate each segment to a time vector

$$\boldsymbol{\xi}^{(n,i)} = \{\bar{\tau}_i + k\frac{\bar{\tau}_{i+1} - \bar{\tau}_i}{t_{i+1} - t_i}\}_{k=0:t_{i+1}-t_i} \tag{7.2}$$

where $\tau_i = \frac{1}{\sum_i T_i}\sum_{i=1}^{N} t_i$ are the average normalized starting time of each segment. The concatenation of all segments therefore forms a normalized time vector.

We train one HMR for each segment using all available trials, to build a left-right hierarchical HMR of the complete model that associates a unit time vector to the full movement sequence.

The model is then ready for generation, and we can synthesize the average movement the regression method presented in Section 6.2 with a unit time vector as input (See in particular Equation 6.11). For consistency, we use the state posterior windowing technique we introduced in section 6.2.5. We estimate the motion feature vector associated to a time value as the weighted sum of the estimations of each segment, weighted by the likelihood of the segment. This constraint guarantees that we synthesize the optimal trajectory, without artifacts at the segment transitions.

7.1.3 Variance Estimation

One of the advantages of HMR is the possibility to estimate, at each time step, the variances over the generated parameters, as discussed in Section 6.4.1. The variances are computed as the sum of the conditional variance of each state, weighted by the squared probability of the state. In our case, the recognition over a unit time vector smoothly interpolates the state probabilities by successively activating the states in the left-right structure.

7.2 Average Performance and Variations

In this section, we study the synthesis of the average movement performance for each participant. The Performance is represented by the trajectories of 3D acceleration captured with an accelerometer placed on the sword.

7.2.1 Consistency and Estimated Variances

Figure 7.1 depicts for participants **T** and **S** the motion trajectory synthesized with the model trained with all trials. The trajectories are surrounded by the the 95% confidence interval over the generated parameters — computed as twice the estimated standard deviation. The standard deviation along the performance is also reported as a shaded curve on the bottom plot. The hierarchical HMR was trained with a single Gaussian and 10 states per segments, yielding a total of 130 states over the entire sequence. We used no relative regularization and an absolute regularization $\lfloor\sigma\rfloor = 1e^{-3}$.

We observe important variations of the confidence intervals over time: the standard deviation is low for the first three segments, and gets larger for the subsequent segments. As the states' variances encode the variability over the training set, the confidence intervals are indeed representing the variability of the performer over time. Investigating the internal values of learned model therefore allows us to analyze the consistency and variability of a performer over a complete movement sequence.

In the same way, the method allows to compare between participants. In this case, it can be surprising that the most expert participant (**T**) presents in some points larger variance than participant **S**. The histogram of the standard deviations for each participant reveals that the expert mover presents a lot more points with small variance ($\sigma < 0.15$)than the student, which might indicate that the teacher is more consistent on a set of key gestures.

7.2.2 Level of Detail

In the previous case, we used only 10 states per segments. The synthesis provides a smooth representation of the movement, that might actually un-

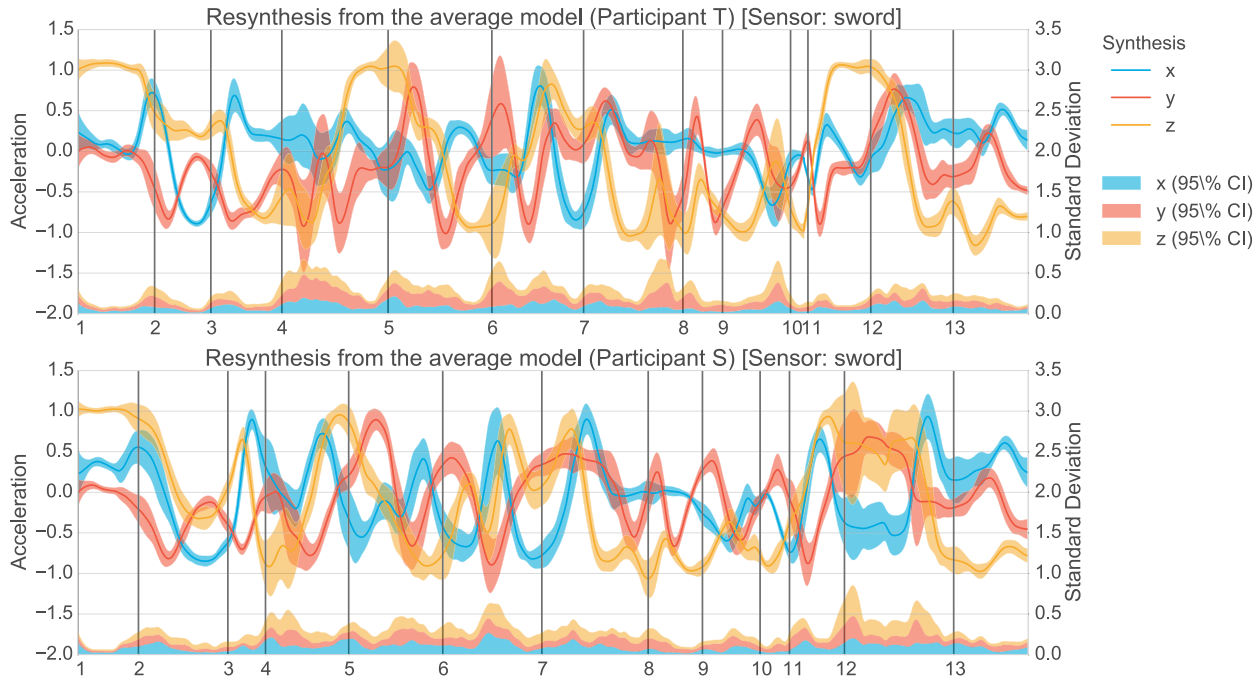


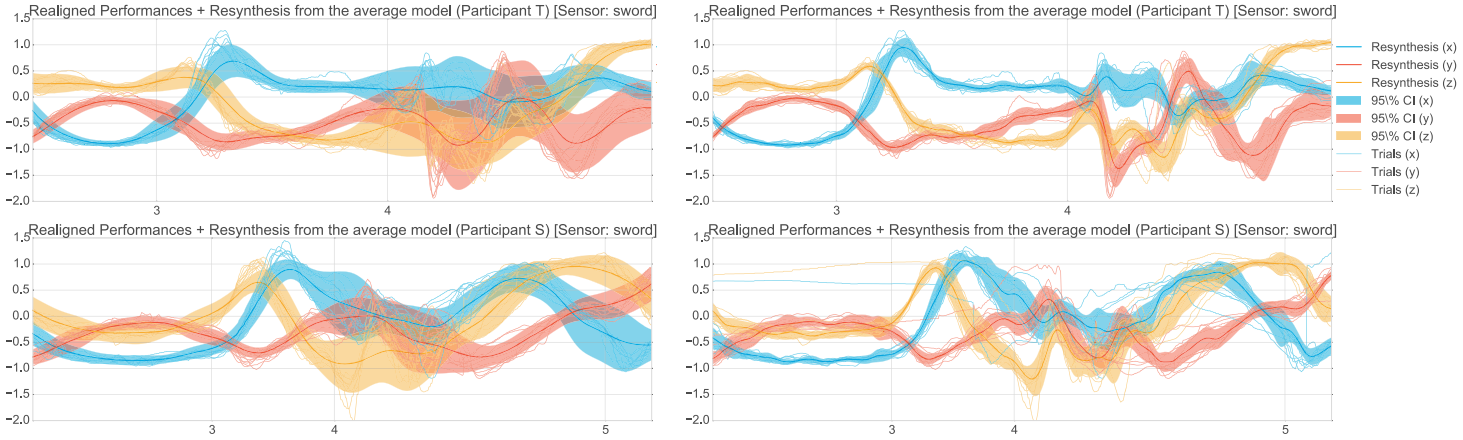
Figure 7.1: Resynthesis from the average model with confidence intervals. The models were trained with the 10 performances of each participant ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = [1e^{-10}, 1e^{-3}]$).

derfit the performance. We can gain a greater level of detail by augmenting the number of hidden states.

We performed the synthesis with models trained with 50 states per segments, yielding a total of 650 states for the full sequence. Figure 7.2 reports a zoom of the synthesized trajectories on the third and fourth segments. The figure depicts the realigned individual performances over the average resynthesis, which confirms the relevance of the estimated confidence intervals that consistently cover most of the trials.

The trajectories reveal a lot more details of the movement: some gestures have sharper curves, and we can observe rapid oscillations patterns. Naturally, increasing the number of state reduces the average variance per state. Nonetheless, in some cases the higher number of states reveals consistent sub-patterns that were underfitted by the 10-state models, and therefore integrated in the variance.

Considering Figure 7.2a, it is obvious that 10 states are insufficient to encode the movement accurately: most acceleration patterns and peaks are clearly undershot. On the contrary, the 50-state model succeeds at reconstructing the acceleration patterns of the fourth segments with reduced undershooting (Figure 7.2b). This higher resolution allows to observe a very specific and reproducible pattern. With only 10 states, this pattern was hidden into the variance, and therefore accounted as noise rather than as a consistent variation. The difference of variability between the teacher and the students is more pronounced. We can observe that the acceleration patterns of participant **T** are more clearly synchronized and have less variability in dynamics than participant **S**.



(a) 10 states ($\lfloor\sigma\rfloor = (1e^{-10}, 1e^{-3})$)

(b) 50 states ($\lfloor\sigma\rfloor = (1e^{-10}, 5e^{-5})$)

Figure 7.2: Detail of the average resynthesis and realigned trials for different numbers of states. The models were trained with a single Gaussian on the 10 performances of each participant. The realigned trials are superimposed to the average resynthesis with confidence intervals.

We studied the internal representation of HMMs learned from several performances of a single participant. HMR allows to represent the set of trials in a single model and to consistently synthesize the average motion trajectories. The estimation of the variances gives insights on performers' consistency over time.

7.3 Synthesizing Personal Variations

We now investigate how the models can be learned jointly from two different performers. Our goal is twofold. First, we aim to validate the ability of HMR to learn from performances with larger variations, and to resynthesize these variations consistently. Second, we discuss how this joint performer model can provide visualization tools to compare performances.

7.3.1 Method

Our methodology follows the protocol of the previous section where we learn a segment-wise mapping between a time vector and a sequence of motion parameters. However, we now learn a single model from all trials of each participant. We extend the method to answer the question: *can we resynthesize each participant's performance from a single model learned from both performers?*.

For this purpose, we add a parameter to the input modality that defines the balance between each participant. The input features are therefore composed of the concatenation of a time vector with a constant parameter vector which value p_{part} differs for each participant. To guarantee a consistent scaling, we chose to associate the respective values $p_{part} = -0.01$

and $p_{part} = 0.01$ to participants **T** and **S**.[1] As before, the output features are composed by the 3D acceleration of the sword's movement. For training, we are associating each motion segment to a time vector whose length averages the lengths of the segments across participants. In order to avoid overfitting, we use a small number of hidden states with important regularization ($N = 10, \lfloor\sigma\rfloor = [1e^{-10}, 1e^{-3}]$).

Once trained, we can use the model for synthesis and allows to interpolate between several behaviors by varying the input time vector and the participant parameter vector.

7.3.2 Results

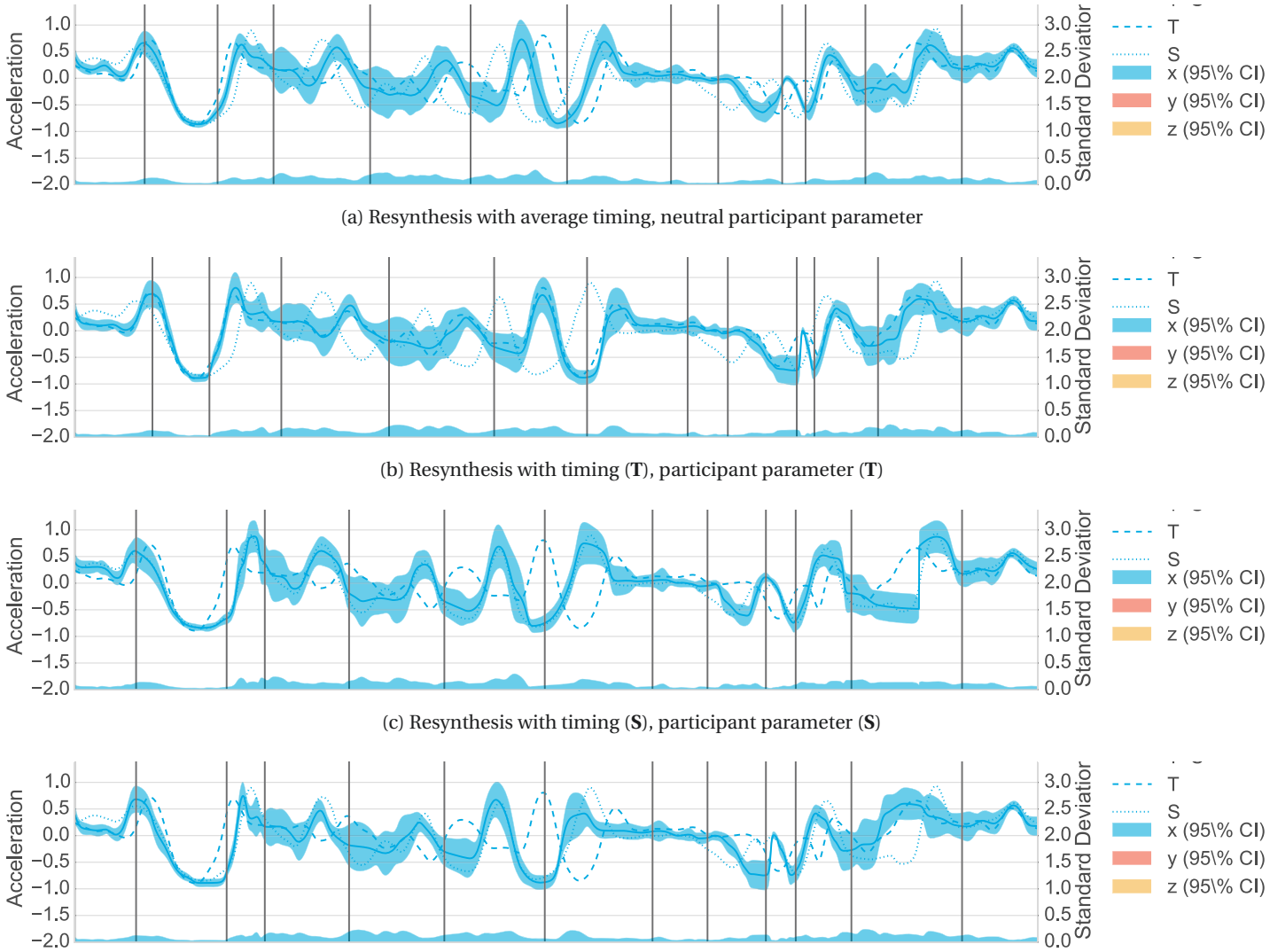
Figure 7.3 summarizes the results obtained for various input time vectors and participant parameters. Although training and synthesis were performed on 3D acceleration, for clarity we only report the acceleration over the X axis. Each plot represents in dashed and dotted lines the average synthesis for each participant with their characteristic timing. The continuous line and colored surface represent the synthesis from the global model with the associated 95% confidence intervals. The vertical bars represent the segmentation associated with the time vector used as input.

Figure 7.3a depicts the movement obtained by synthesizing with the average timing and a neutral participant parameter ($p_{part} = 0.$). We observe that the average timing correctly distributes each pattern of acceleration equally between the average performance of each participant. Moreover, the shape of the acceleration signal represents an intermediate shape between each participant's acceleration patterns. Therefore, the synthesis of the average behavior seems consistently interpolate between the synthesis of each participant. It highlights high consistency on certain gestures (for example, segment 6), while averaging with high variance when the participants exhibit different behaviors (segments 5, 12).

Figures 7.3b and 7.3c depict the syntheses obtained by using the respective timing and participant parameters of participant **T** and **S**. Even though the quality of the synthesis is lower than using participant-specific models, we observe that the generated movement differs from the average model and tends to reproduce the acceleration pattern of the given participants. This result is confirmed by the RMS errors between the syntheses obtained with the global and the participant-specific models. Whatever the timing used for synthesis, the global RMS error is systematically lower between the global synthesis with a given participant parameter and the synthesis obtained with the target participant's model.

The interest of building such a participant-parametrized model is the possibility of interpolating between participants both on timing and dynamics. This is illustrated in Figure 7.3d where we used the timing of participant **S** combined with a participant parameter corresponding to performer **T**.

1 Note that scaling is of major importance for this application: larger values of the participant parameter can lead to convergence errors or inconsistent syntheses.



(a) Resynthesis with average timing, neutral participant parameter

(b) Resynthesis with timing (**T**), participant parameter (**T**)

(c) Resynthesis with timing (**S**), participant parameter (**S**)

(d) Resynthesis with timing (**S**), participant parameter (**T**)

Figure 7.3: Synthesized trajectories using the participant-parametrized global model. Each trajectory is generated with a different combination of the time vector and the participant parameter.

We proposed to learn a global HMR model of the performance, parametrized by the participant. HMR effectively allows to reproduce the specific behaviors of each participants, and consistently interpolates between the trajectories.

7.4 Vocalizing Movement

We propose to study the relationship between vocalizations and movement performance. During the recording session, we asked participant **T** to vocalize along her movement performance of the Tai Chi sequence. In this section, we study the synthesis of the average trajectories of sound descriptors in relationship to the movement performance.

7.4.1 Synthesizing Sound Descriptors Trajectories

We recorded 8 performances of participants **T** where we invited her to vocalize along her movement. We didn't give any additional instruction, and she was free to choose the timing and type of vocalization to use. We recorded the audio from a DPA microphone synchronously to the acceleration sign from the 3 sensors.

METHOD The sound is described using Loudness and Spectral Centroid descriptors, resampled at 100Hz to match the acceleration signal. We propose to synthesize the average descriptor trajectory using a mapping learned with 8 vocalized performances.

We train a hierarchical HMR model on the 13 segments of the full sequence, with a left-right topology similar to the approach presented in Section 7.1. The input movement is represented using the 3D acceleration from three sensors, and the sound feature vector are the concatenation of the Loudness and Spectral Centroid.

We propose to synthesize the average descriptor trajectory of the full performance as follows. We use the average movement performance synthesized as described in Section 7.2, using 50 states per segment. Then, we synthesize the associated descriptor trajectory using hierarchical HMR. The resulting trajectory is therefore estimated from the average movement performance using the mapping trained over all trials of the participant.

RESULTS Figure 7.4 reports the synthesized average descriptor trajectories of Loudness and Spectral Centroid. The mapping was performed with a hierarchical HMR with 10 states per segments. The individual trajectories of each trial — realigned using the movement performance, — are plotted in thin blue lines.

We observe that the performer does not vocalize continuously, but several gestures seem to be supported by vocalizations, as indicated by the loudness peaks. The vocalization therefore seems to be related to the performed movements. Moreover, the performer presents a high consistency in the timing of the vocalizations across trials. In the figure, all individual trials are realigned to the average movement performance with the alignment method presented in Section 5.2.1. The synchrony of the loudness peaks across trials testifies of the consistency of the timing of the vocalization with specific movement patterns.

The observation of the synthesized average descriptor trajectory gives insights on the consistency of the vocalizations. It appears that the vocalizations at 1, 12, 15 and 20 seconds are highly consistent across trials, as the synthesis of the loudness presents few artifacts. Other gestures present more variations, such as the vocalization between 5 and 10 seconds. In this case, the vocal sound presents a lot of variations in timing and dynamics across trials, and the average synthesis fails at reconstructing a clear loudness pattern.

The difference is even more striking on the synthesis of the spectral centroid (Figure 7.4, bottom). Several patterns, e.g. the first pattern between

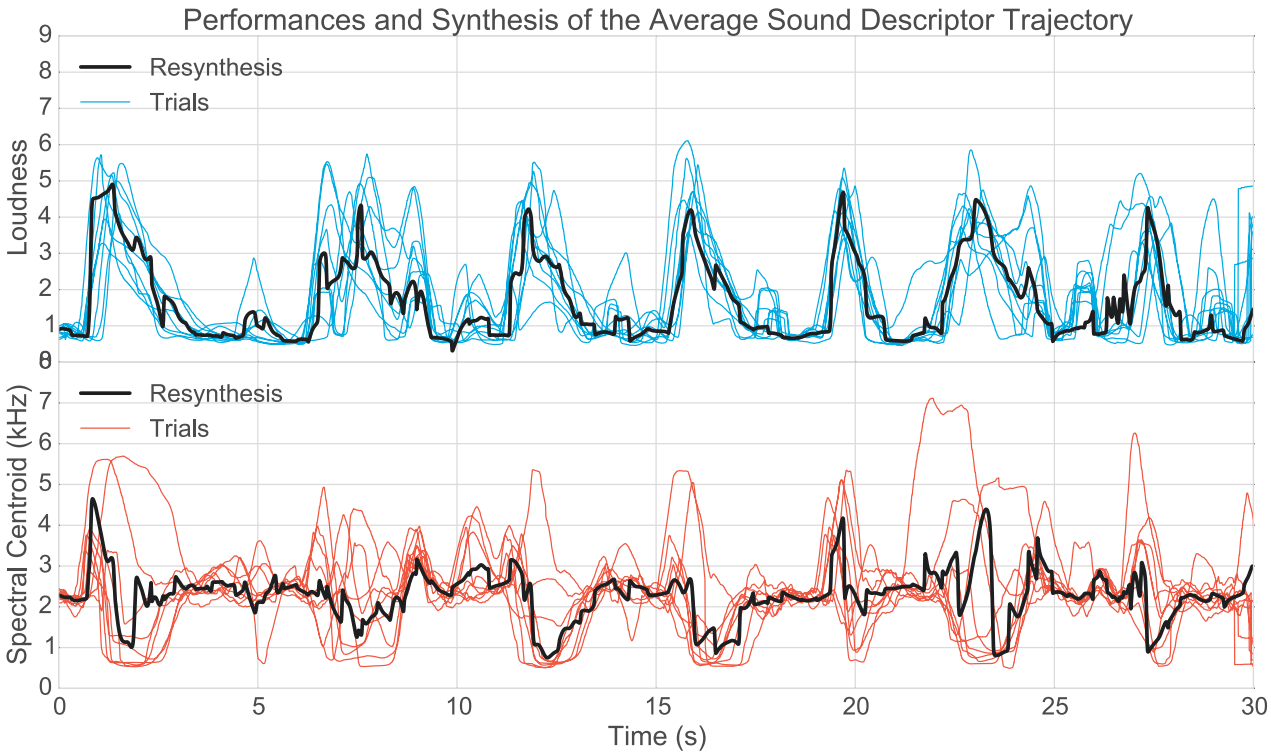


Figure 7.4: Synthesis of the descriptor trajectories from the average movement performance. The descriptors are estimated using hierarchical HMR with 10 states per segments, a single Gaussian per state ($N = 10$, $M = 1$, $\lfloor\sigma\rfloor = [1e^{-3}, 1e^{-10}]$)

1 and 5 seconds, present different trajectories, that give evidence of variations in the vocalization strategy. This correlates with the observation of the audio-visual recording. We observed that, while the performer consistently performed vocalizations with given gestures, she could use both voiced and unvoiced vocalizations for sonifying the same movement pattern.

7.4.2 Cross-Modal Analysis of Vocalized Movements

We propose to combine the methods for synthesizing the average trajectories of both motion and sound for analyzing the relationship between the gestures and the vocalizations.

Figure 7.5 reports a plot of the synthesized motion trajectories from the model of participant **T**. The bottom part of the plot puts in relationship the synthesized average loudness trajectory along with the variance of the movement over time.

We observe that high loudness of the vocalization often correlates with small values of variance estimated over the generated motion parameters. For most of the vocalizations, the motion variance decreases along the vocalizations. This correlation highlights that the performer consistently synchronizes the vocalizations with gestures that are reproducible with high consistency.

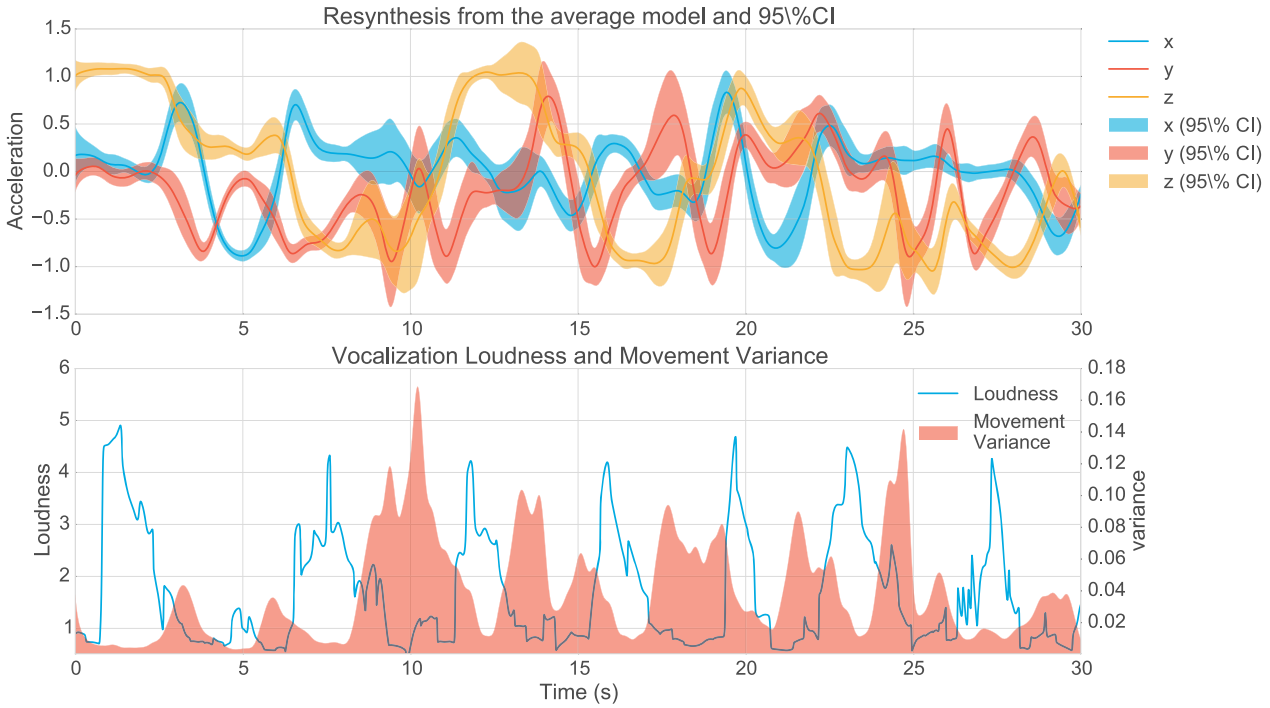


Figure 7.5: Loudness of Vocalizations and Movement Variance. The top plot represents the synthesized average movement, the bottom plot jointly represents the vocalization loudness and the variance estimated on the movement performance.

> We proposed to investigate vocalizations performed while moving. Our method uses HMR for synthesizing the average trajectories of loudness and spectral centroid associated with the average movement performance. This analysis highlights a high consistency in the timing of the vocalizations, that often occur during the gestures that are most consistently executed.

7.5 Summary and Contributions

This section investigated movement analysis using a generation approach. We proposed a method for synthesizing the motion performances using Hidden Markov Regression (HMR). The method allows to synthesize both the average trajectory and the associated variances given a set of trials from a performer, which gives insights into the consistency of a movement performance across trials, and possibly across participants. Parametrizing the HMR model over performers allows to interpolate in timing and dynamics between the behavior of different performers, and validates the ability of HMR to model some variations in movement performance. Finally, we proposed a method for investigating the cross-modal relationships between motion and sound in vocalized performance. Such analysis highlights a consistent timing of the vocalizations, that often occur along the gestures that are most consistently executed.

8

Playing Sound Textures

In this chapter, we present a generic system for crafting gestural strategies for interacting with sound textures. The system exploits the parameter generation approach presented in Chapter 6. It uses Gaussian Mixture Regression (GMR) to learn the mapping between motion and sound descriptors. The system has applications in sound design for interactive systems such as gaming, music, or rehabilitation.

With listening as a starting point, we investigate how the perceptive attributes of sound textures shape gesture design, through two experiments. First, we study novice users' strategies for associating motion to environmental sounds in the framework of a public installation with an end-user Mapping-by-Demonstration (MbD) system. Second, we report on a user study investigating if interactive sonification based on such a strategy can improve the reproduction of particular gestures.

OUTLINE We start by motivating the approach with respect to the related work in corpus-based sound synthesis, perception studies, and Human-Computer Interaction (Section 8.1). In Section 8.2, we present a general overview of the application's architecture and implementation. Finally, we detail two instantiations of the system. Section 8.3 presents an end-user installation presented at SIGGRAPH 2014 Studio, and Section 8.4 reports on a controlled experiment investigating the use of sound feedback for the reproduction of three-dimensional gestures.

8.1 Background and Motivation

Our system exploits a MbD approach to the design of gestural control strategies of sound textures. This system is grounded in a body of research along several themes. First, it relates to studies in sound perception that aim to understand how people associate movements and sounds. We aim to study if MbD helps novice users implementing personal control strategies relating to their embodiment of particular sound textures. Second, we want to explore the use of interactive motion sonification with sound texture to support movement learning.

8.1.1 Body Responses to Sound stimuli

Recent theories of perception and cognition emphasize the role of motor behavior in sound perception. In particular, recent results in neuroscience support this "motor theory of perception", suggesting that motor images are integral to sound perception (Kohler et al., 2002). Specific aspects of this auditory-motor coupling have already been highlighted, such as the influence of music on motor timing in tapping tasks (Large, 2000). Yet, the understanding of how people associate movements with sounds remains underexplored.

Several research groups started to investigate such phenomena. Godøy et al. (2006b) studied the gestures associated with music playing in 'Air Instruments', highlighting the coupling between between instrumental performance and a 'mimicking' gestural paradigm. In Godøy et al. (2006a), the same authors investigate 'sound tracings' by asking people to move along sounds — without further instruction, — and suggest that causality might be an important factor in the motor representation of sounds. In sound perception, Lemaitre et al. (2010) showed that both expertise and identification of the sound causal source are determining factors in sound classification. Caramiaux et al. (2010b) further investigated gestural responses to sound stimuli, showing through cross-modal analysis that short abstract sounds often result in more consistent control strategies correlating speed and loudness.

Caramiaux et al. (2014b) reports a user study where participants were asked to perform gestures while listening to sounds. The sounds are split into two group: non-transformed sounds the cause of which is clearly identified (e.g. "pouring cereal into a bowl"), and transformed versions of the same sounds that prevent their identification. Caramiaux et al. show that the induced movement are either metaphorical when the cause of the sound is identified — participants mimic the action that cause the sounds, — or people tend to 'trace' the temporal evolution of the timbral characteristics of the sounds when its source cannot be identified. These results echo Gaver's taxonomy opposing *musical listening*, that focuses on the acoustic qualities, to *everyday listening* where causality play an essential role (Gaver, 1993).

Caramiaux et al. (2014b) focused on ecological sounds that instrinsically refer to human actions. Their observations highlight that participants consistently attempt to imitate the action causing the sounds, but their corporeal expression of the action itself is highly variable. In this application, we consider environmental sounds which origin is easily identified, but that do not necessarily refer to elementary human actions. Our goal is to investigate how people intrinsically relate environmental sounds to hand movements, and what is the influence of the shift from *sound-accompanying gestures* to *sound-producing gestures* (Godøy et al., 2006b) on such gesture-sound associations in a Mapping-by-Demonstration framework.

8.1.2 Controlling Environmental Sounds

Our proposal is motivated by other applications in sound design, that relate to gaming, motion picture, or music. Environmental sounds have become essential in both musical and non-musical domains. *Musique Concrète* and the developments of field recording pushed in-situ audio recording to the front of the experimental music scene. In the visual industries, both motion picture and gaming require the development of sound environments to support the action onscreen. Often, these application require to generate environmental sound textures that constantly evolve to avoid repetition, but which characteristics must be controlled accurately to support the interaction. In motion picture, *Foley Art* is still one of the most expressive ways to create sonic environments, through the manipulation of — sometimes unrelated — objects. However, foley art can be tedious: producing water sounds may require to bring in the studio a water tank to record a soundtrack synchronously to visual events, which is both impractical and time consuming. Gestural interaction might offer an additional tool in Foley artists' palette to expressively control the sound variations.

Recent advances in sample-based sound synthesis, in particular Corpus-Based Concatenative Synthesis (CBCS) (Schwarz, 2007), make it possible to explore and render high-quality synthesis of sound textures (see for example Schwarz (2011) for a comprehensive review). CBCS is a data-driven approach to sound synthesis that relies on a database of annotated sounds (a *corpus*) containing a large number of audio segmented and their associated description. Their description aggregates both low-level descriptors (spectral descriptors, perceptual descriptors) and possibly high level descriptors (categories, symbolic attributes). The technique can synthesize sound according to a target sequence of such descriptors. This sequence might be provided either as input audio (in the case of audio *mosaicing* (Zils and Pachet, 2001; Aucouturier and Pachet, 2006; Tremblay and Schwarz, 2010)), or through direct exploration of a descriptor space (Schwarz et al., 2006). CBCS has many applications in music as an instrument (Schwarz, 2012), or in sound design for the control of textures (Schwarz and Caramiaux, 2014), and can be coupled with novel tangible interfaces (Savary et al., 2013). Yet, the expressive power of gestural interaction has not been fully explored.

We draw upon Diemo Schwarz's *CataRT*[1] (Schwarz et al., 2006; Schwarz, 2004), a complete sound analysis/synthesis framework for Corpus-Based Concatenative Synthesis. CataRT provides tools for visualizing an interacting with a corpus through a 2D user interface that represents each *grain* (or segment) in a descriptor space. The 2D interface provides a direct access to sound from descriptors, allowing to expressively *navigate* within sounds with a perceptually-relevant control. However, the interface is limited to the simultaneous control of 2 dimensions, and does not provide and intrinsic mechanism for playing with multiple corpora at the same time.

We propose a system for gesture control of corpus-based concatenative synthesis. It uses 3D hand gestures to interact with the sound corpus,

1 http://ismm.ircam.fr/catart/

where the control strategies are specified by demonstration. Our system allows users to define several gestures, possibly associated to several sound corpora, with a continuous recognition mechanism that enables them to navigate between several textures.

8.2 Overview of the System

This section describes the principle and implementation of our generic system for gestural control of sound textures. We start by outlining the workflow of the system, and we then detail the sound design, demonstration, and performance components.

8.2.1 General Workflow

An overview of the complete system is depicted in Figure 8.1. The process starts with the design of the sound examples. The sounds are directly synthesized using corpus-based concatenative synthesis with CataRT's graphical user interface (see Section 8.2.2). Several corpora can be used, and several sound examples can be generated from each corpus of sounds. Once recorded, the sounds examples can be used as basis for creating a control strategy for a given corpus.

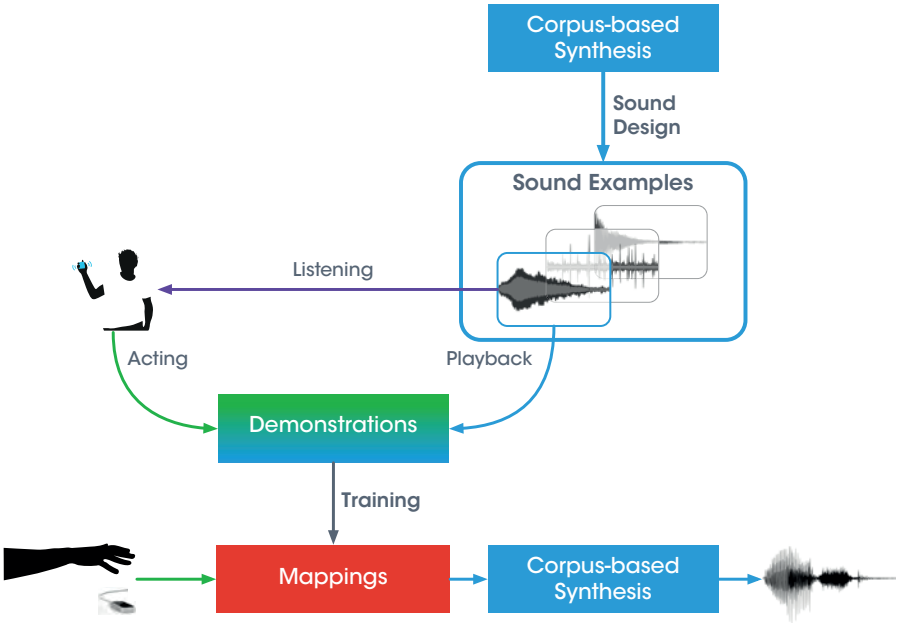


Figure 8.1: Overview of the System

The user must then perform a gesture along the sound example to act out the motion-sound relationship. The joint recording of the motion and sound parameters[2] forms the *demonstration*. The system is independent

2 Along this section, sound parameters refer to the sound descriptors used as input to the descriptor-driven concatenative sound synthesis.

to the type of input device used for motion capture. In the following applications, we use the Leap Motion[3] for capturing hand movements.

We train a Gaussian Mixture Regression (GMR) model over motion and sound parameter sequences. In performance, we use GMR to predict the sound parameters associated with an input hand movement, that drive the concatenative synthesis engine accordingly.

The subsequent sections detail the main components of this process: the initial sound design with CataRT, the recording of the demonstrations, and the performance phase.

8.2.2 Sound Design

We design the sound examples using CataRT, according to the process described in Figure 8.2.

We first build a corpus by analyzing a set of sound files. Each audio recording is segmented using onset detection, and we compute a set of sound descriptors for each segment. In the following experiments, we describe the sound with both perceptual descriptors (e.g. Loudness) and spectral descriptors (e.g. Spectral Centroid).

CataRT displays the corpus by spreading the points associated with each segment over a 2D space defined by two descriptors. Figure 8.2 depicts a corpus of wind sounds with Spectral Centroid as abscissa, Loudness as ordinate, and a color coding based on periodicity.

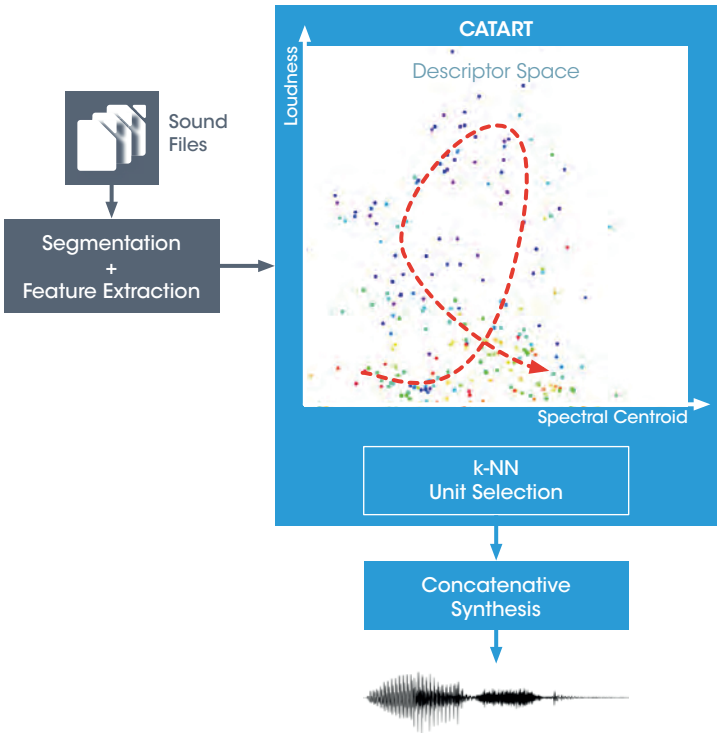


Figure 8.2: Design of the sound examples with CataRT.

3 http://leapmotion.com/

With this graphical representation, we can explore the corpus from the description itself, by navigating across the 2D space with a mouse. For synthesis, CataRT uses a k-Nearest-Neighbor (k-NN) search to find the sound segments which description is the closest to the input XY position in the descriptor space. The sound is synthesized by concatenating the target segments. Several parameters act on the textural 'qualities' of the synthesis. The temporal envelope of each segment (duration, attack and release time) and the overlap between successive segments, combined with additional filtering, specify the density of the grains.

This allows us to design the sound examples intuitively, by drawing trajectories in the descriptor space (red dashed arrow in Figure 8.2). For each example, we record the temporal trajectory of the descriptors. The sounds can then be played back by streaming the descriptors to the concatenative synthesis engine.

MUBU IMPLEMENTATION We use an alternative implementation of CataRT based on *MuBu*[4] (Schnell et al., 2009). MuBu is a generic container designed to store and process multimodal data such as audio, motion tracking data, sound descriptors, markers, etc. The MuBu software package provides a set of tools for visualization (`imubu`), recording and playing utilities, and sound synthesis engines (additive, granular and concatenative syntheses).

Both the segmentation and feature extraction are computed using PiPo[5] (*Plugin Interface for Processing objects*), a set of real-time processing utilities associated with the MuBu framework. We use a MuBu container to store both the audio content and the descriptor sequences. The visualization of the corpus and the 2D mouse control are realized with the scatterplot interface of `imubu`.

CORRESPONDENCE The advantage of using CataRT for designing the sound examples are twofold. First, it allows to control the sound parameter variations very accurately, and according to perceptually-relevant sound descriptors. Second, it allows to maximize the sound *correspondence*, as discussed in Section 3.4. Using CataRT both for the initial sound design and during interaction ensures that the sounds are generated in the same way for *demonstration* and *performance*.

8.2.3 Demonstration

Once the set of sound examples is built, creating the demonstrations is straightforward. The playback of the example is generated in real-time by replaying the sound parameters trajectories into the concatenative engine.

Once users have imagined and practiced the gesture they wish to associate with the sound, they can record it while listening to the sound. The motion parameters are recorded synchronously to the sound parameters.

4 http://ismm.ircam.fr/mubu/
5 http://ismm.ircam.fr/pipo/

Then, we learn the mapping between motion and sound parameters using Gaussian Mixture Regression (GMR). Expert users can adjust the parameters of the model: number of components, regularization. Otherwise, the training is hidden to novice users and is automatically performed at the end of the demonstration. In our experiment, training the model never takes more than a few seconds for training sequences inferior to a minute.

8.2.4 Performance

The Performance phase starts as soon as the training finishes. The movement features extracted from a live movement are streamed to the GMR model, that estimates the associated sound parameters. These descriptors are used to control the concatenative sound synthesis, as depicted in Figure 8.3.

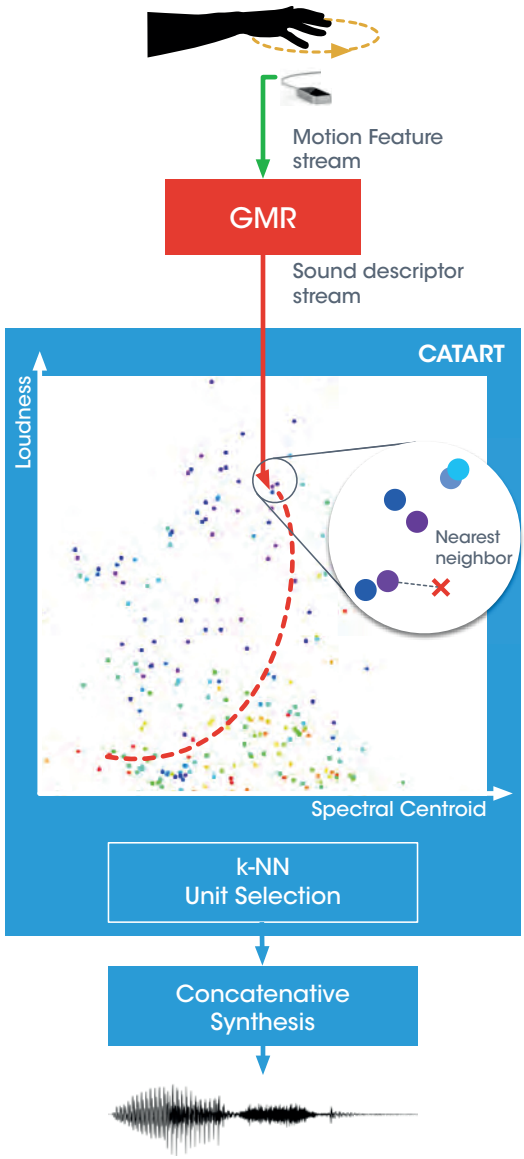


Figure 8.3: Performance: Mapping with GMR and k-NN based unit selection.

The sound synthesis is similar to that of the initial sound design: the estimated position in the descriptor space is used to select the set of grains to play using a k-NN search within the corpus. The selected grains are concatenated according to the synthesis parameters (envelope and overlap between grains).

8.3 Siggraph'14 Installation

We developed a public installation based on the generic system presented above. We demonstrated the application at SIGGRAPH 2014 Studio and Emerging Technologies in Vancouver, Canada. The installation ran for five days in the demonstration area of the conference. SIGGRAPH'14 gathered a wide public of about 14.000 artists, researchers and professionals in the fields of computer graphics, motion picture, and interactive techniques. We designed the system to be usable for a wide public, without technical knowledge of machine learning nor musical interactive systems.

In this section, we detail the specificities of the system presented during the conference and we report qualitative observations and feedback from the attendees.

8.3.1 Interaction Flow

The installation is an adaptation of the general system presented in the previous section. To fit the target audience of novice users, we simplified the process of sound design and training, and complemented the scenarios with several new features.

We simplified the demonstration phrase, by predefining the sound corpora and sound examples. We chose a set of eight corpora of sounds from the *DIRTI–Dirty Tangible Interface*[6] project by Matthieu Savary, Florence Massin and Denis Pellerin (User Studio[7]); Diemo Schwarz (Ircam); and Roland Cahen (ENSCI–Les Ateliers). The sound corpora were designed by Roland Cahen.[8]

We created a single sound example for each corpus, by drawing in CataRT a closed-loop trajectory varying in Loudness and Spectral Centroid. The final set of sound examples is available for listening online, along with a screenshot of the application[9].

The interaction flow is illustrated in Figure 8.4. First, users can select and listen to a sound example. Once imagined, they can record a gesture with a single hand while listening to the sound example.

The gesture must be recorded periodically during 5 loops of the sound example. For training, we discard the first and last examples of the recording to ensure that motion and sound are consistent and correctly synchronized. There are two advantages in recording several executions of the ges-

6 http://www.smallab.org/dirti/
7 http://www.userstudio.fr/
8 http://roland.cahen.pagesperso-orange.fr/
9 http://julesfrancoise.com/phdthesis/#siggraphinstall

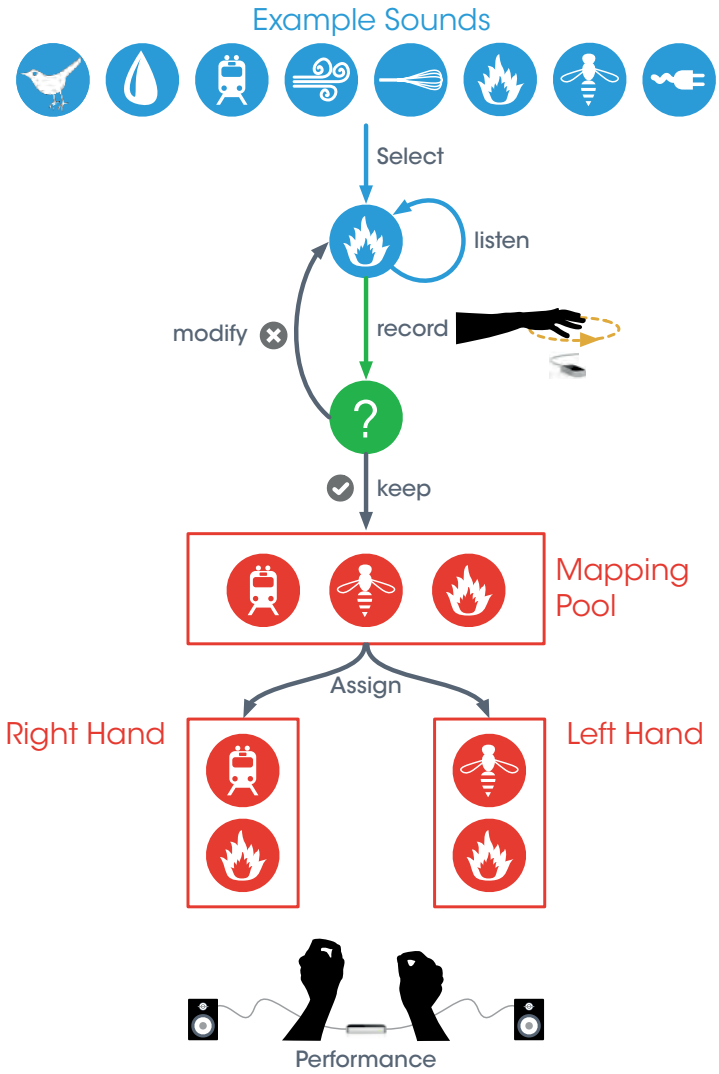


Figure 8.4: Interaction Flow of the Siggraph 2014 Studio Installation.

ture along with the sound: it makes the training more robust by integrating noise over several executions, and it optionally allows to define several variations of the gesture to control the same sound.

The end of the recording triggers the training, that automatically activates an evaluation mode that allows the user to directly interact with the sound according to the relationship defined by her example. Depending on the satisfaction with their design, users can either record the demonstration again, or add the trained model to a *pool* that aggregates the learned mappings.

Users can record up to eight demonstrations with various sound corpora. In *Performance* mode, users can choose to assign any mapping from the *pool* to either or both of their hands. For consistency, the motion parameters are symmetric between the left and right hand, and both hands are generating sound independently. Using bimanual input allows to mix between several sounds and control strategies simultaneously, yielding rich sound environments.

We now describe the gesture capture and description, and we present our strategies for handling multiple corpora simultaneously in performance.

8.3.2 Movement Capture and Description

We use the Leap Motion commercial sensor for tracking the movement of the hand. For this purpose, we developed a Max external connecting to the Leap Motion Skeletal tracking SDK *V2.0* (See Appendix A.2.7). The object provides the full skeleton model of the hand, with several additional parameters.

As this application is dedicated to novice users, we selected parameters describing the global movement of the hand rather than using the skeleton tracking at the finger level. We compute eight motion features:

SPEED is the speed of the center of the hand along the X, Y and Z axes.

ORIENTATION is composed by the X, Y, Z coordinates of the normal vector to the palm.

GRAB STRENGTH continuously estimates the closing of the hand (0=flat, 1=closed fist).

SHAKINESS describes if the hand is shaking periodically. This feature is computed from the sum of the power spectra of the speed over each axis. It is defined as the product of the normalized spectral centroid by the spectral energy of the movement. Therefore, it activates when the hand has periodic movement with enough energy.

8.3.3 Interaction with Multiple Corpora

We enriched the interaction by adding the possibility to superpose several mappings during performance. We developed a specific strategy for handling multiple mappings and corpora, which is described in Figure 8.5 for the case of two mappings.

We train one GMR for each demonstration associating a gesture and a trajectory of sound descriptors. In performance, the GMRs run in parallel: the input movement is streamed to each model, that estimates both the likelihood of the gesture and its associated sound parameters. The sound descriptors estimated by each model are streamed to concatenative sound synthesis engine, the gain of which is determined using the posterior likelihood of the associated model. This way, instead of performing a strict classification of the different gestures, we continuously overlap and mix the different sounds.

This process allows to develop rich control strategies, considering the variety of features that are captured from the input movement. For example, one can choose to design similar gestures for two sound corpora, to be able to smoothly interpolate or superpose two sound textures to create a mixed environment. On the contrary, defining gestures with very different qual-

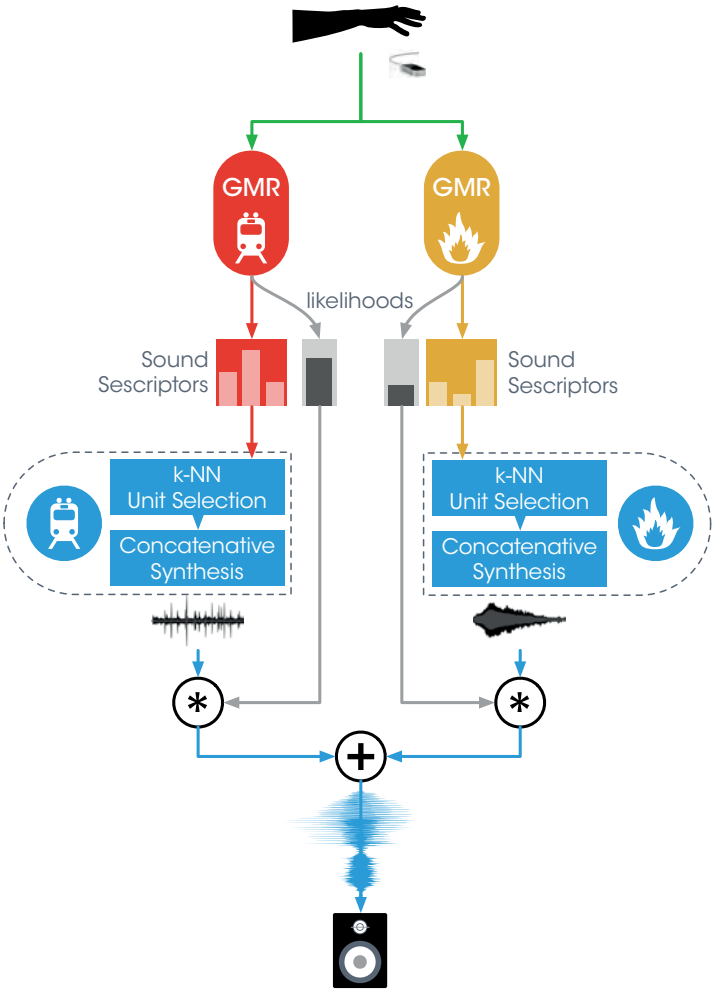


Figure 8.5: Combining mappings using posterior likelihood mixing

ities results in a reduced overlap between the models' region of support, which amounts to classification.

8.3.4 Textural and Rhythmic Sound Synthesis

We implemented an option for changing the parameters of the concatenative sound synthesis. In the default mode, sound segments are triggered periodically every 50 milliseconds, with an attack time of 5 milliseconds, and a release time of half the grain duration. With such settings, the grains are continuously chained with overlap: the sound synthesis is continuous and has textural qualities. A 'groove' mode allowed to change the synthesis parameters to a period of 150 milliseconds (4 segments per second) with 100 milliseconds duration and 80 milliseconds release time. In 'groove' mode, the segments are therefore rhythmically triggered rather than continuously chained.

It is interesting to note that while the parameter mapping is not altered — the relationship between motion parameters and sound descriptors remains unchanged, — changes in the sound synthesis have a great impact

on the perceived relationship between motion and sound. By introducing a mismatch in the *correspondence* between demonstration and performance sounds, we widen the gap between the experience of moving while listening and that of moving to control the sound synthesis. As a result, the 'groove' mode significantly impacts on the movements of the user in interaction, often inducing a rhythmic synchronization of the gestures to the sound.

8.3.5 Observations and Feedback

The presentation in conference gave us the opportunity to experiment in the wild with an end-user Mapping-by-Demonstration approach. Over the five days of presentation, we estimate that several hundred attendees could experiment the system. We report here several findings arising from qualitative observations and participants' feedback.

END USER MAPPING-BY-DEMONSTRATION

Globally, we observed that novice participants apprehended the system easily. Several people highlighted the intuitiveness of the system for creating personal mappings.

Often, participants were able to reproduce the sound example by performing the demonstration gesture again, and most participants reported that the system consistently reflected their intended relationship. Several factors could lead to inability to reproduce the original sound: tracking errors during the recording (e.g. finger movements, losses of tracking), and inconsistent demonstration across the five examples used for training. With particular gesture designs, the system could also extrapolate the mapping strategy to explore new zones of the sound corpus.

STRATEGIES

We observed a wide variety of gesture strategies, combining speed and orientation of the hands, and occasionally involving subtle finger movements. The diversity of strategies did not allow us to identify clear strategies for associating gestures to environmental sound textures, but we observed that most participants tend to preserve an energy continuum between the gesture (speed) and the sound (loudness). Also, several participants associated the time scale of sounds to physiological constraints, by linking slow evolutions of the textures to hand movements and quick and impulsive sounds to rapid finger gestures.

The strategies seem to differ according to the sound corpus. Three corpora were very textural and induced smooth hand movements: *wind*, *water* and *fire*. The corpora of *bird* and *electric* sounds contained mostly short impulsive sounds, that induced movements with the same granularity, and were adapted to a use in the 'groove' mode. Other were less used, such as *train* or *kitchen*. While this latter corpus contains several action-related sounds, the mapping technique did not allow for controlling each segment through a metaphor of its sound-producing action, and was less appreciated. Globally, the continuity of the corpus was an important factor.

TECHNOLOGICAL LIMITATIONS

The experiment highlighted several technological limitations, mostly due to the Leap Motion device used for continuous hand tracking. The commercial software extracts a full skeleton of the hand from a stereo infrared camera. The system is therefore sensitive to lighting conditions, and can be unstable under lighting with non-negligible infrared wavelengths (e.g. halogen lights).

Another limitation is the size of the interaction zone, that can be constraining for two-hand interaction. Optical occlusions arise when the hands are above each other and can result in tracking errors. This problem was critical in our application where demonstrations are performed with a single hand, whereas both hands can be used in performance. Gestures designed for the complete interaction zone with a single hand can hardly be reproduced with both hands in performance.

Other limitations are due to the choice of motion descriptors. For simplicity, we chose to use only high-level features of the hand movement. Several participants created gestures involving subtle finger movements that were hardly represented by the hand features and lead to inconsistencies. In some cases, however, finger movements impacted the hand speed and orientation and lead to relatively consistent strategies. Along the interaction, participants understood which features were tracked and redesigned their movements accordingly.

8.3.6 Discussion

We observed that participants globally appreciated the interaction with the system and reported its intuitiveness and entertainment values. We noted a large variety of strategies for associating hand movements to environmental sounds, often reflecting an energy continuum between motion and sound, and a consistency of the time scales between gestures and sounds. More specific studies would be required to understand in more detail how such associations are made.

We observed an interesting sensori-motor adaptation phenomenon during the public experiment. We noticed that most participants were able to reproduce the sound example (used for demonstration), after a few attempts to reproduce the gestures. It seems that for such mid-air gestures, that do not intrinsically provide haptic feedback, the addition of interactive sonification could help participants reach a better consistency for reproducing their own gestures.

8.4 Experiment: Continuous Sonification of Hand Motion for Gesture Learning

The presentation at SIGGRAPH gave insights on the usability of the system and allowed to identify relevant gestures and types of sounds. These findings informed the design of an experiment that aims to investigate the influence of continuous sonification on the performance of specific gestures. Echoing the concept of Mapping-by-Demonstration, we conducted

a user study where participants were asked to imitate gestures from an audio-visual recording, with or without continuous sonification. This section reports the design, analysis methods, and the main findings of the experiment.

8.4.1 Related Work: Learning Gestures with Visual/Sonic Guides

GESTURE LEARNING WITH VISUAL FEEDBACK

Human-Computer Interaction (HCI) is concerned with the development of efficient and expressive forms of interaction. As tangible interfaces become ubiquitous, users must constantly adapt to new input modalities that require to learn and master particular gesture techniques. Although studies support that user-defined gestures are easier to recall and execute (Nacenta et al., 2013; Wobbrock et al., 2009), they can be challenging due to misconceptions of the recognizer's abilities (Oh and Findlater, 2013). For robust recognition, predefined gesture sets are the most widespread, which led to the development of a thread of HCI concerned with providing users with novel means to learn such gesture sets. For mouse and keyboard interaction, several methods have already been proposed to improve task efficiency, for example through visual and auditory feedback for speeding-up the memorization of hotkeys (Grossman et al., 2007), or through particular menu layouts such as the marking menu (Kurtenbach and Buxton, 1994).

Similar approaches have been proposed for tangible interaction, using onscreen visuals to guide gesture interaction. Octopocus (Bau and Mackay, 2008) is a dynamic guide combining feedforward and feedback mechanism for help users to "learn, execute and remember gesture sets". Octopocus continuously displays the possible gestures from a given pose, with their associated actions, along with the state of the recognition system. Bau and Mackay (2008) show that dynamic guides significantly reduce the input time compared with help menus and hierarchical marking menus. Appert and Zhai (2009) highlighted how using strokes as shorcut was as efficient as hotkeys while decreasing the learning time and improving long-term recall.

Most approaches focus on multi-touch gestures on 2-dimensional surfaces devices that can provide co-located and situated visual feedback. We consider the case of 3-dimensional mid-air gestures that have become essential in full-body interaction, for example with large displays (Nancel et al., 2011). In this case, situated visual feedback is difficult to implement and might add a heavy cognitive load. As an alternative, we investigate sound as a rich feedback modality, aiming to study if and how interactive sonification can help gesture learning.

According to Anderson and Bischof (2013), learnability involves two factors: the cognitive mapping between gestures and actions (associative learning), and the ability to perform a gesture. As it is already well-known that auditory feedback can help associative learning (Gaver, 1986), we focus only on the performance of arbitrary gestures.

MOTOR LEARNING WITH AUDITORY FEEDBACK

While vision has long been the primary feedback modality in motor learning, a recent thread of research in sensori-motor learning research investigates audition as an extrinsic feedback modality.[10] Auditory perception has a lower cognitive load and faster response than visual feedback (Baram and Miller, 2007), yet it can carry rich information content. Using interactive sonification for movement learning has many applications such as HCI (Oh et al., 2013), motor rehabilitation (Robertson et al., 2009), or sport where it aims to improve performance and accuracy (Effenberg et al., 2011).

Often, direct mapping strategies are used for sonifying physical quantities. Reviewing 179 publications related to motion sonification, Dubus and Bresin (2013) highlight that simple sound synthesis prevails — many works use pure tones varying in pitch and amplitude, — and that pitch and panning are the most frequently used sound parameters, often directly driven by position or velocity. These simple feedback have two major shortcomings. First, Sigrist et al. (2013) underlines that *descriptive* feedback might be less efficient that *prescriptive* feedback, and argue for error-related sonification strategies. Sigrist et al. highlight the difficulty of error sonification in applications such as sport where specifying the 'target' movement might be difficult. Second, the use of basic sound synthesis can be ineffective in mid- to long-term learning tasks: practicing with unpleasant sound feedback can even degrade performance. At the other end of the spectrum, several authors propose to use music to support exercising and rehabilitation. In this case, music mostly supports learning through engagement and rhythmic interaction rather than continuously informs on movement execution. Some authors argue for the use of ecological sonification in the case of sport Effenberg et al. (2011), especially in conjunction with virtual reality environments.

GOAL OF THE EXPERIMENT

We investigate if auditory feedback on continuous movements can improve motor learning. We report an exploratory study applying Mapping-by-Demonstration to movement learning, that uses the system presented in the previous section to sonify movements with environmental sounds. The study focuses on reproducing arbitrary hand gestures performed by another person. The participants are trained by moving along a video recording of the experimenter's movement, with additional sonification. Then, the participants are asked to record three series of executions of the gestures. We evaluate two conditions: reproduction with the interactive sonification, where the mapping is adapted to the participant's movement during practice; and a silent condition where no feedback is given to the participant.

10 See in particular the LEGOS project at ircam: http://legos.ircam.fr

8.4.2 Method

APPARATUS The experiment ran on a Macbook Pro running Mac OSX with a Leap Motion for continuous hand tracking. The setup of the experiment is schematized in Figure 8.6.

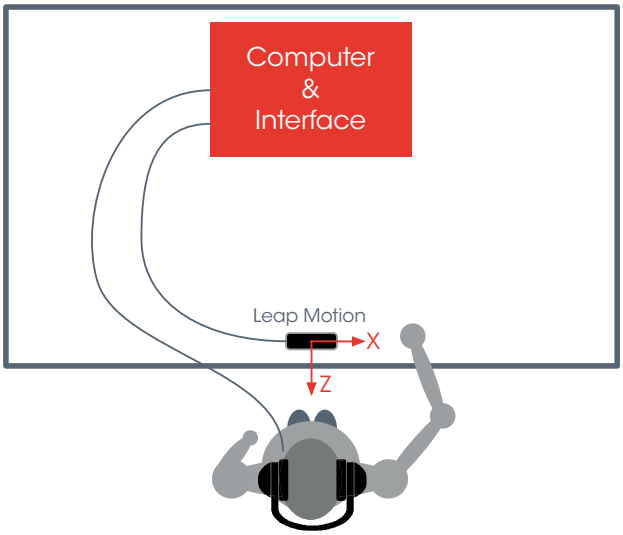
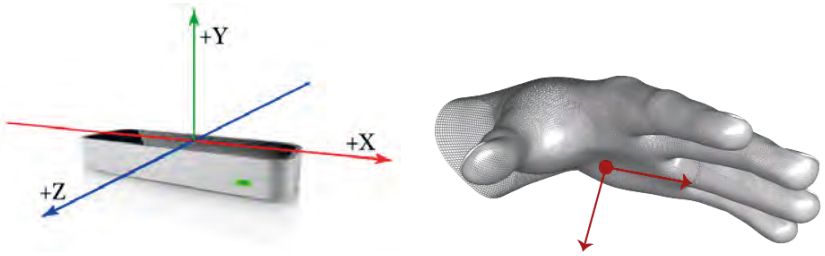


Figure 8.6: Schema of the experimental Setup. Hand movements are tracked using the Leap Motion. The computer is used for gesture acquisition, mapping, and sound synthesis.

We used only global features of the hand movement: the 3-dimensional speed in Cartesian coordinates, and the orientation vector given by the normal to the palm (see Figure 8.7). The device was placed before the computer with the Z axis oriented towards the participant. To ensure the stability of the tracking, participants were first explained the size of the interaction area, and were instructed to keep the hand open with the fingers slightly spread. We asked the participants to stand up during the recordings, to ensure a correct elevation of the hand and to limit arm fatigue. The tracking data was acquired in Max using the `leapmotion.mxo` external[11] exploiting the Leap Motion Skeletal Tracking SDK V2.0 and resampled at 100 Hz.



(a) Coordinate System of the Leap motion

(b) Tangent and normal palm vectors

Figure 8.7: Coordinate System of the Leapmotion

11 See Appendix A.2.7

The interface was developed with Cycling'74 Max 6 and integrated movement acquisition, mapping, sound synthesis, and the GUI elements necessary to the experiment. A screenshot of the software used in the experiment is depicted in Figure 8.8.

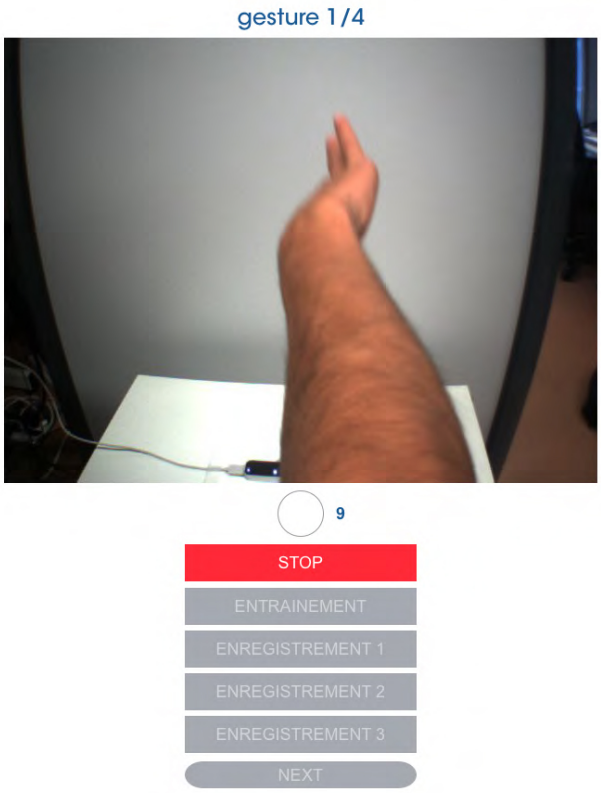


Figure 8.8: Screenshot of the interface used in the Experiment

The sonification was perfomed using the system described in Section 8.2. The mapping used Gaussian Mixture Regression (GMR) based on the `mubu.gmr` object from the Max implementation of the XMM library. The synthesis was performed by the MuBu implementation of CataRT.

TASKS The experiment focuses on reproducing the gestures of another person. We designed 4 gestures associated to sounds, and we made an audio-visual recording of one execution of each gesture. This recording served as the reference gesture (or 'target') that the participant had to reproduce. The task was formulated as follows: "You will reproduce the observed gesture as accurately as possible and record several repetitions, trying to be as consistent as possible". To facilitate the understanding of the gesture's shape and dynamics, it was videotaped from the viewpoint of the performer — to maximize the *correspondence* between the participants viewpoint and the demonstration. The demonstration consisted of an audio-visual demonstration of the gesture: in both conditions, the gesture was presented with the video and the associated sound feedback. The participants were instructed to practice while watching the recording, and were then asked to record several executions of the learned gesture.

GESTURES AND SOUNDS The design of the gestures, of the sounds, and of their association was informed by the observations from the presentation at SIGGRAPH'14. We selected two sound corpora: *wind* and *water*, that we identified as the most continuous sets presenting clear timbral variations. We designed two sounds for each corpus using CataRT. Each sound was generated from a continuous closed-loop trajectory in the two-dimensional descriptor space defined by *Loudness* and *Spectral Centroid*. The sounds were 3 to 4 seconds long, and presented slow variations of Loudness and Spectral Centroid. We iterated the design of the gesture set over several pilot studies. Each gesture of the final design was created according to the dynamics of the sound, with a relative congruence of the movement's energy with the loudness. We attempted to design equally difficult gestures combining both three-dimensional speed variations and smooth changes in orientation. The audio-visual demonstrations were created as follows:

1. design of the sound using CataRT.
2. design of the gesture while listening to the sound.
3. Learning of the Mapping using GMR.
4. Recording of the final demonstration with the interactive sonification based on the learned mapping.

The final set of gestures and sound demonstrations is available online as supplementary material.[12]

PROCEDURE Each participant performed the reproduction of the 4 gestures. Feedback was the primary factor with two levels: *Sonified* (**S**) and *No Feedback* (**N**). The experiment was composed of 4 main blocks, one for each gesture to reproduce. Each participant performed two gestures with the sound feedback (**S**) and the two other gestures without any feedback (**N**). The 6 possible associations between gestures and conditions were balanced across participants, and the order of presentation of the gestures was randomized, under the constraint that two gestures with the same condition cannot follow each other. Half of the participants started with condition **S**, while the other half started without feedback (**N**).

Figure 8.9 illustrates the detailed procedure for one gesture. The first block (**D**) is the *Demonstration*: the audio-visual recording is played 10 times without interruption. The participants can already move to mimic the demonstration gesture. In the second block (**P**), participants must *practice* while watching the demonstration, again 10 times. They are informed that their gesture is recorded, and that they must try to imitate the gesture as accurately as possible. We keep the 5 last executions of the gesture for training a Gaussian Mixture Regression model with the user's gestures — excluding executions with tracking errors. The experiment continues with 3 recording blocks containing each a series of 10 executions

12 http://julesfrancoise.com/phdthesis/#leapexp

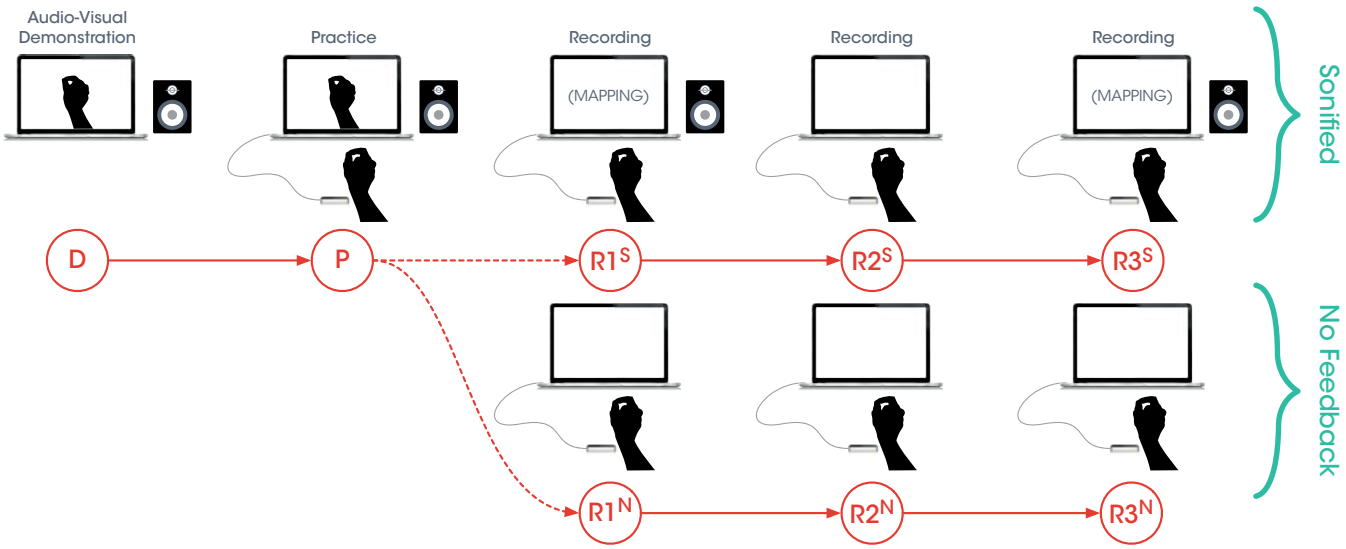


Figure 8.9: Protocol of the experiment (S=Sonified, N=Silent)

(**R1–R3**). In condition **N**, all 3 recording blocks are performed without feedback. In condition **S**, the movement is sonified in the first and last recording blocks (**R1**$^{\mathbf{S}}$, **R3**$^{\mathbf{S}}$), and no feedback is provided in the middle recording block (**R2**$^{\mathbf{S}}$). A 30 seconds break is imposed between each block to avoid fatigue.

In each recording block, the participants must perform 10 executions of the gesture. Each execution must be followed by a pause, and a beep is triggered when the hand is still to indicate that the next execution can be performed. To ensure the gathering of correct executions, we log the hand tracking errors from the Leap Motion. Such errors occur when the participant moves outside the interaction zone of the device, or when the skeleton model fails to fit the hand of the participant. The trial is discarded if such an error occurs and the participant has to continue until 10 correct executions are recorded. No other indication of the state of the tracking is provided[13].

At the end of the experiment, the participants were invited to fill a questionnaire, that included a set of questions related to the difficulty of the task and the influence of the sonification. The questionnaire included 3 questions to assess on a 5-point Likert Scale. The total duration of the experiment was between 20 and 30 minutes.

PARTICIPANTS We recruited 12 participants, among whom 6 were female, both researchers from Ircam and undergraduate students. The age of the participants ranged from 19 to 47 ($mean$ = 26.9, SD = 9.2). All participants were right-handed, the experiment was exclusively performed with the right hand. Most participant reported an experience with movement practice: 7 reported regular sport practice, and 4 had experience in music computing.

13 In the pilot studies we implemented either visual or auditory alarms, that were found disturbing and were later removed for the final experiment.

ANALYSIS We aim to investigate the participants' ability to reproduce the demonstrated gesture with the help of interactive sound feedback. Therefore, we are interested in tracking the accuracy of the participants over time. We propose to study how the distance between participants' executions and the reference gesture evolves along trials.

Deriving metrics between gestures is not straightforward, due to the difficulty of identifying relevant invariants between similar gestures. We choose to evaluate the accuracy of gesture reproduction using distances between multidimensional motion parameters trajectories of speed and orientation. In particular, we compute a distance between each trial of the participants and the reference gesture performed by the demonstrator.

For analysis, we select the set of correct trials from each participant. Each trial is automatically segmented using energy thresholds to select the region of interest where the gesture is performed. Finally, the selected executions are manually analyzed to discard failed executions, possible tracking errors, and segmentation errors.

The choice of a distance between gestures must meet several constraints: it must measure both variations in timing and amplitude of the parameters, but must remain invariant to possible time delays. We rule out the euclidean distance that does not allow any possible delay or temporal variation between the target and test gestures. Dynamic Time Warping (DTW) has been used as an alternative to account for temporal variations between the performances but is not appropriate in our case, where timing remains important.

We propose the use of a constrained correlation distance, defined as the minimal euclidean distance between two gestures with varying delays:

$$d(\boldsymbol{g}_{ref}, \boldsymbol{g}_{test}) = \min_{\tau \in [-\Delta, \Delta]} \frac{1}{T_\tau} \sum_{t=1}^{T_\tau} \|\boldsymbol{g}_{ref}(t) - \boldsymbol{g}_{test}(t-\tau)\| \tag{8.1}$$

where $\|\boldsymbol{a} - \boldsymbol{b}\|$ is the euclidean distance between frames $\boldsymbol{a}$ and $\boldsymbol{b}$, T_τ is the number of overlapping frames between the two sequences with delay τ, Δ is the maximum authorized delay. The statistics are computed as follows: for each participant, each gesture, and each block, we compute the average distance between all trials of the block and the reference recording of the experimenter's gesture. This raises for each participant 4 measures in the practice block **P**, and 12 measures in recording blocks. In order to smooth the differences between participants, we focus on tracking the evolution of the distance in the recording block with respect to the average distance in the practice block. All distances of the recording blocks are divided by the average distance in the practice block.

8.4.3 Findings

We now report both quantitative and qualitative results that emerge respectively from the analysis of the participants' gestures, and from observations and participants answers to the questionnaire.

CONSISTENCY OF REPRODUCTION

First, we compare the distance to the reference across participants and gestures, for the three recording phases and the feedback conditions. As illustrated in Figure 8.10, we identify a difference between the feedback conditions. Without feedback (condition **N**), the distance tends to increase along the recording blocks, while it is more stable in the sonified condition **S**.

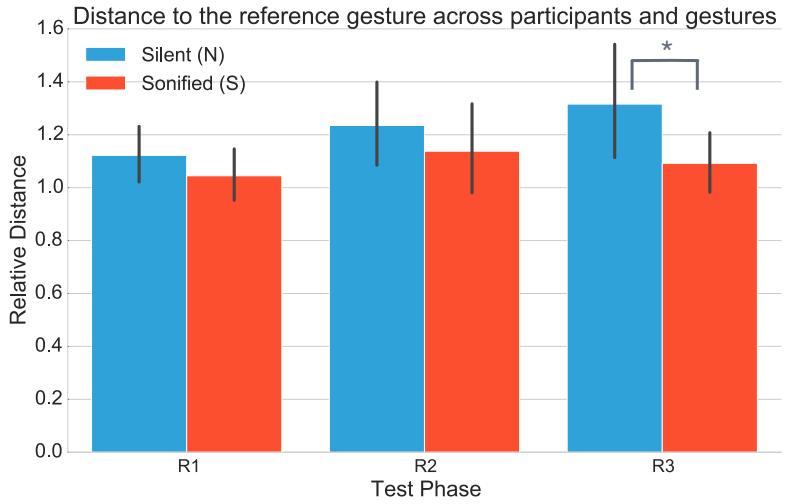


Figure 8.10: Distance to the reference gesture for each phase of the experiment, averaged over participants and gestures. The figure presents a bar plot representing the median and 95% confidence interval of the sliding euclidean distance.

No significant difference between feedback conditions was found for phases **R1** and **R2**. With a two-way ANOVA, we found an effect of the feedback condition in phase **R3**, with a lower distance in condition **S** (mean=1.32, SD=0.50) than in condition **N** (mean=1.09, SD=0.28). The significance was however borderline with a medium effect size ($F_{1,36} = 3.61$, $p = 0.06$, $\eta^2 = 0.07$). The effect of the gesture was also significant ($F_{3,36} = 2.36$, $p = 0.08$, $\eta^2 = 0.03$).

These results are associated with the distance computed on the XYZ speed features only. Note that significance is borderline when the distance combines both speed and orientation features, and that distances computed over orientation only do not show a significant difference between feedback conditions. In the following, we consider the results obtained on XYZ speed trajectories.

VARIABILITY ACROSS GESTURES

It appears that gestures performed with the interactive sonification lead to lower distance to the reference gesture. Nonetheless, there is an important variability between the four base gestures. We now detail the results obtained for each gesture.

Figure 8.11 details the distances between trials and reference gestures across participants and trials, for each of the four base gestures. We observe that the results significantly vary between the four proposed gestures

and associated sounds. We tested for statistical significance using non-parametric Mann-Whitney U tests for each gesture, each recording block, across participants[14].

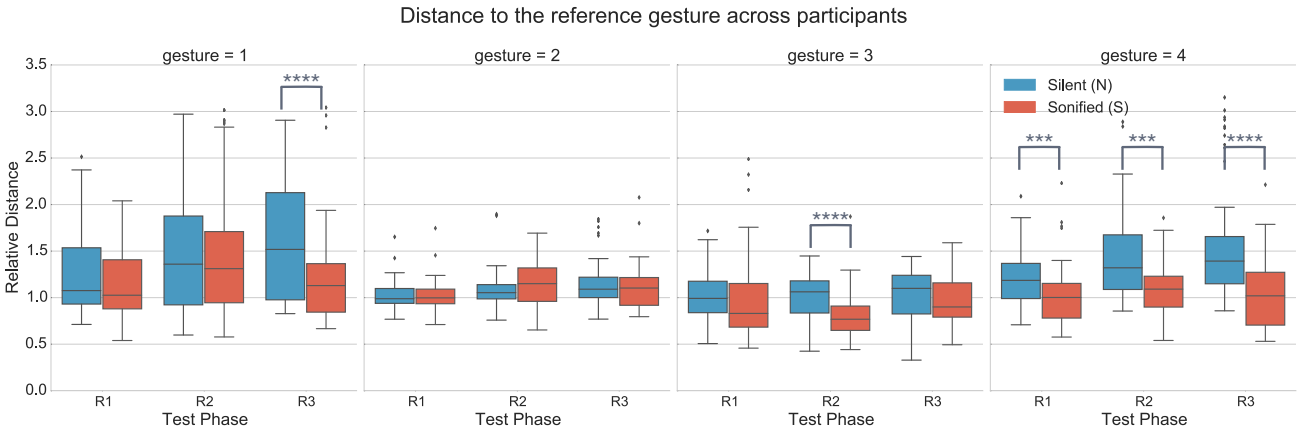


Figure 8.11: Distance to the reference gesture for each gesture and each phase of the experiment, across participants and trials. The figure presents a box plot representing the median, 1st and 3rd quartiles of the sliding euclidean distance.

Gestures 1 and 4 exhibit clear differences: in condition **N**, the distance to the reference gesture tends to increase along recording phases. On the contrary, the distance remains approximately constant across recording phases when the gestures are performed with the help of interactive sonification. In this case, the distance is slightly higher in recording phase **R2**, where the sonification is absent. For gesture 1, we found a significant difference in the distances in recording phase 3 where the medians for conditions **N** and **S** are respectively 1.52 and 1.13 (The mean ranks of Group **S** and Group **N** were 45 and 64, respectively; U = 912, Z = -3.18, $p < 0.001$, r = 0.31). For gesture 4, the distance was found lower in condition **S** than in condition **N** for all three recording phases, under $p < 0.01$. This difference is all the more important in phase 4, where the median distance in conditions **N** and **S** is respectively 1.39 and 1.02 (The mean ranks of Group **S** and Group **N** were 38 and 69, respectively; U = 643, Z = -5.06, $p < 0.001$, r = 0.48).

For gesture 2, no statistically significant difference was found between feedback conditions, and the distance slightly augments across recording blocks in both conditions. Similarly, for gesture 3, if it seems that the distance is lower with the sonification from the first recording phase, we only found a significant difference between feedback conditions in phase **R2**, where the sonification is not present (The mean ranks of Group **S** and Group **N** were 43 and 70, respectively; U = 901, Z = -4.14, $p < 0.001$, r = 0.38). Interestingly, it seems that gesture 2 was the hardest to reproduce, as

14 While the global statistics are computed using averaged distance per participant and block, such averaging does not provide sufficient data for statistical tests when investigating each gesture independently. The test by gestures therefore take into account all trials of all participants. In this case, the distances are not normally distributed, and the hypotheses for parametric test do not hold.

it combined complex orientations of the palm with a circular movement, while gesture 3 was found the easiest.

> Some gestures present a clear difference when performed with the interactive sonification. With the interactive sound feedback, the distance augments, indicating that the gesture drifts from the demonstration. For gestures 2 and 4, gestures performed with the interactive sonification results in a more stable behavior across trials.

SOURCES OF ERROR We now seek to understand what are the differences in gesture performance between feedback conditions for gestures 2 and 4. Specifically, we aim to identify whether the difference observed in the global distance are due to dynamic or timing variations. For analysis, we segmented the gestures using energy thresholds, and the resulting gestures were manually selected to remove outliers and failed trials.

Figure 8.12 presents the average duration of the trials across participants for the four base gestures, relatively to the average duration in the first recording phase.

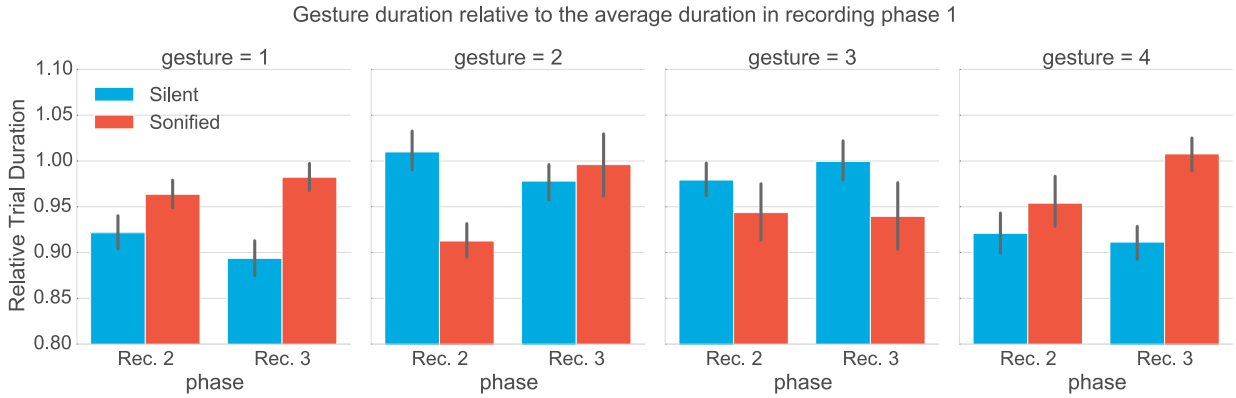


Figure 8.12: Average gesture duration across participants and trials, for each gesture and each phase of the experiment. The figure presents a bar plot representing the mean and 95% confidence interval of the duration of each gesture relative to the first recording phase.

Gestures 1 and 4, which show lower distances in the sonified condition, present less variations in duration along time when performed with interactive sonification. In condition **S**, the average duration in phase **R3** is almost equal to the average duration in the first recording phase **R2**. Participants without sound feedback tend to accelerate more critically that participants with the interactive sonification. The comparison with other metrics for measuring the distance between gestures highlighted that the changes in timing are indeed the most critical factor in the observe differences on the distance.

> Interactive sonification results in a lower distance to the reference gesture along trials, that is mostly due to a more consistent timing when gestures are performed the sound feedback.

Participants' Appreciation of the Sonification

At the end of the experiment, we invited the participants to fill a questionnaire asking for their experience in music, movement practice, and containing several questions on the experiment.

Participants were asked to answer on a 5-point Likert scale which rating were annotated for each question, and could elaborate their answer verbally. The first two question focused on the difficulty to *memorize* and *reproduce* the reference gestures, respectively. The third question concerned the effect of the interactive sound feedback for reproducing the reference gesture more accurately. The results of the Likert scale questions are reported in Figure 8.13.

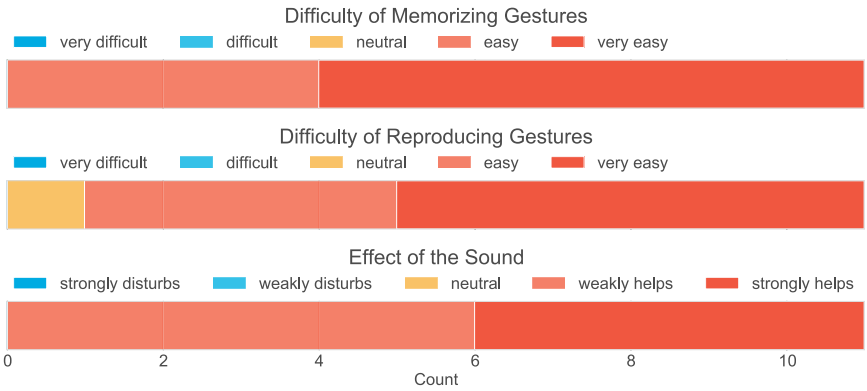


Figure 8.13: 5-Point Likert scores to the questionnaires items.

Most participants reported that gestures were *very easy* or *easy* to memorize. Several participants highlighted that the 30 seconds break imposed between recording phases made memorization more difficult. A single participant reported a *neutral* difficulty of reproducing gestures, others found it either *easy* or *very easy*. However, several participants noted that it was sometimes difficult to reproduce all characteristics of the gesture, in particular the combination of orientation and speed variations.

By observing the participants passing the experiment, we identified a large variety of strategies for imitating the demonstrated gestures. All participants have a different re-embodiment of the observed gesture, which results in a great variability in the performance of the gestures. Although the gestures were designed to be simple, and were filmed in a first-person setup, it seems that there is a non-trivial correspondence between the experimenter's movement and the participant's movement.

The third question was about the interactive sonification: participants were asked to evaluate whether the interactive sound feedback helped reproducing the reference gesture more accurately. Globally, participants reported that sound supports a more accurate reproduction of the gestures. Some participants reported that sound was helpful for keeping a consistent timing, and a participant evoked imagining the sound when performing the gesture without the interactive sonification, to keep a better regularity. Another participant reported that the sound helped adjusting the angles. More specific comments highlighted some drawbacks of the sonification strategies. With corpus-based sound synthesis, some grains that are close

in the loudness-centroid descriptor space can be perceptually different if their timbre varies more critically. Some participants reported that triggering grains that had not been listened in the reference demonstration could be disturbing. Another participant pointed out that it was difficult to understand how the feedback explicitly encoded the error to the reference gesture, and consequently, how to improve from the interactive sonification.

8.4.4 Discussion

We reported on a controlled experiment investigating gesture imitation with interactive sound feedback. The measure of the evolution of the distance between the participants' gestures and the demonstration along trials highlights that interactive sonification results in a stable distance to the reference. In absence of feedback, participants' gestures tend to drift over time. The analysis of the durations of the gestures underlines that timing is more influenced by the sonification than dynamics.

LIMITATIONS While the effect size of the interactive sonification is small in our experiment, the results are promising considering the specificities of the experiment design. As a matter of fact, several aspects limit the efficiency of the sonification.

First, if descriptor trajectories are continuous, corpus-based sound synthesis might introduce discontinuities in the perceived timbre, that were found disturbing.

Second, if the participants were informed that their gestures were recorded in the practice phase, they were not informed that these gestures were used to adapt the mapping. We observed that it could be difficult for some participants to synchronize with the video and to manage to perform high quality imitation while watching the demonstration. Therefore, the examples used to adapt the mapping did not necessarily have a sufficient quality to optimize the mapping. A direct improvement would invite users to record a few examples they found 'satisfying' after sufficient practice.

Third, the experiment was very constrained in that participants were not allowed to freely explore the sound feedback before recording the gesture reproductions. This design choice came from a will to guarantee the same conditions of practice in both feedback conditions. We would gain insights into the ability to learn movement with sound feedback by letting participants practice and explore the sound environment in more details before recording the final executions.

Acknowledgments
We thank Olivier Chapuis for fruitful discussions on the experiment design. Part of this experiment was realized for a student project in collaboration with Sylvain Hanneton at Université Paris-Descartes. We wish to thank Sylvain Hanneton, Caroline Barbot and Amélie Briaucourt for their support and involvement in the project. With partial support of the ANR Legos project.

8.5 Summary and Contributions

In this chapter, we proposed a generic system for crafting gesture-based control strategies for interacting with sound textures. The system implements a Mapping-by-Demonstration approach using Gaussian Mixture Regression (GMR) to design the relationship between hand gestures and trajectories of sound descriptors that control corpus-based concatenative sound synthesis. The gestures can be defined while listening to sound examples, allowing novice users to design their own associations between motion and sound.

We applied the system to two different contexts: an interactive installation presented at SIGGRAPH'14 Studio, and a controlled experiment focusing on gesture learning. Observations and qualitative feedback from the installation's attendees highlighted the ease of use of the system, and underlined a wide variety of strategies for associating hand gestures to environmental sounds. The controlled experiment, investigating whether interactive sonification could help improve motor performance in a gesture imitation task, show that the gestures are performed more accurately with the sound feedback. Motion sonification, in this case, mostly supports regular timing in movement execution.

9

Motion-Sound Interaction through Vocalization

In this chapter, we consider voice and gestures as primary material for interaction design in a Mapping-by-Demonstration framework. We present a generic system that uses jointly performed gestures and vocalizations to design the relationship between motion and sound. The system allows users to continuously control the synthesis of vocalizations from continuous movements, using a joint probabilistic model of motion and sound.

The system uses the Hidden Markov Regression (HMR) method introduced in Chapter 6 to learn the relationship between a sequence of motion parameters and a sequence of descriptors representing the vocal sound. In performance, we use Hidden Markov Regression (HMR) to generate a stream of vocal descriptors from continuous movements. We present two applications in sonic interaction design: a public installation based on game scenario, and a sonification system for supporting movement practice.

OUTLINE In Section 9.1, we give an overview of the related work investigating the use of vocalization in conjunction with movement in the contexts of movement practice and sound computing. Then, we present the system for interactive vocalization (Section 9.2). In Section 9.3, we report on an interactive installation presented at the SIGGRAPH'14 conference, that implements a game based on gestural and vocal imitations. Section 9.4 presents the results of an exploratory workshop with dancers that drew from expert movements and vocalizations for the sonification of Laban effort factors.

9.1 Related Work: Vocalization in Sound and Movement Practice

While speech recognition and analysis has been a privileged field in computational modeling and machine learning for several years, it often considers speech in a disembodied form — e.g. mediated by a smartphone or computer. Novel approaches tend to integrate non-verbal communica-

tion and interaction as a central research question in Human-Computer Interaction (HCI). Gesture is now considered a primary interaction modality. Vocalization — i.e. the vocal production non-verbal sounds, — is starting to be investigated in modeling communities, even though it is ubiquitous in human conversation, particularly to communicate emotion: laugh, hesitation, acknowledgement, onomatopeia, vocal imitations, etc.

In this thesis, we consider several use-cases using vocalization as primary material for movement interaction design. As a matter of fact, several domains of movement practice such as dance make an extensive use of vocalization, in particular to support movement pedagogy. At the other end of the spectrum, vocalization has become a tool for acting out sonic interactions in music and sonic interaction design. This section briefly reviews the use of vocalization in these domains.

9.1.1 Vocalization in Movement Practice

Dance practice and pedagogy make an extensive use of vocalization to support movement expression. Reflecting upon her practice of dance teaching in classrooms, Moira Logan notes:

> Sound and movement are elements to be explored together. The familiar human sounds of sighing, laughing, or crying are coupled with movement, amplified and exaggerated until they become larger than life. After more vocalization the children are ready to compose their own vocal score which becomes the accompaniment for a dance based on gesture.
>
> (Logan (1984))

In dance movement therapy, several somatic movement approaches include vocalization as a tool for expression. For example, *Dance & Voice Movement Integration*, developed by Patricia Bardi, or Paul Newham's *Voice Movement Therapy* include breath, touch and voice to facilitate movement expression. Often, vocalization is used as a vector of emotion:

> Any sound that a person offers can be incorporated in the group experience. As people become more comfortable with vocalization, the therapist can encourage sounds that are expressive of particular feelings. [...] In combination with movements, the sound can increase the range of expression and communication. (Best et al. (1998))

However, using the voice for movement expression is not limited to novice practitioners. According to Irmagard Bartenieff, one of the major historical figures of Laban Movement Analysis, "Movement rides on the flow of Breath" (Bartenieff, 1980). The use of Breath in LMA allows the human body to access a large palette of movements, by supporting the phrasing of movement and the full body shaping. Bartenieff emphasizes the crucial role of vocalization as an amplification of Breath in achieving a fluidity and expressivity in movement.

Many choreographers integrate vocal sounds in rehearsal and practice with performers. Kirsh et al. (2009), investigating the choreographic process of Wayne McGregor with the Random Dance company, underline the importance of multimodality for communicating choreographic ideas. The choreographer combines verbal descriptions, prosody and intonation, gesture, touch, as well as vocalization. Kirsh et al. note that vocalization contributes to specifying timing and rhythm, but also movement dynamics and 'quality'; however, without necessarily the same interpretation among dancers. Vass-Rhee comments on the work of choreographer William Forsythe:

> At work in the studio, Forsythe vocalizes constantly, generating aural images of his own or others' movement. This common practice of onomatopoeic or ideophonic vocal reflection, in which vocal gestures (like "pow" and "bling") describe object attributes (such as size, constitution, position, movement, or the temporal structure of events), reflects the embodied nature of sound perception. (Vass-Rhee (2010a))

Forsythe pushed the use of vocal sounds even further by integrating them into his creations, for example to create *intermodal counterpoints* where "Vocalizations engender movements and vice versa as the dancers conduct each other or, responding to direction, translate others' actions and sounds into a visuo-sonic composition of artificial human and animal languages." (Vass-Rhee, 2010b).[1] Forsythe's concept of 'Breath Score' encompasses vocalizations for synchrony among dancer and movement dynamics, that reflect and support the action onstage.

9.1.2 Vocalization in Music and Sound Design

VOCAL IMITATIONS Sound designers often face the issue of searching for specific sounds in massive database. Commonly, This search is supported by meta-information associated to the audio recordings. An alternative approach consists in providing an example sound that serves as a prototype for a query in the database — namely, query-by-example.

Voice constitutes an efficient way of producing sounds, and several authors proposed to use vocal imitations for query by content. Esling and Agon (2013) propose multiobjective time series matching for query-by-example in sound databases, where the query can be provided as a vocal imitation. The method returns multiple propositions optimizing the temporal evolution of multiple timbral properties. The approach of Cartwright and Pardo (2014) relies on a distance that weights the various features according to user ratings of the sounds at each step of the process.

To validate the effectiveness of such vocal imitations, Lemaitre and Rocchesso (2014) conducted a study comparing verbal descriptions of sounds

1 An illustration of Forsythe vocalization process for *One Flat Thing:Reproduced* is available one the *Synchronous Objects* website: http://synchronousobjects.osu.edu/content.html#/TheDance

with their vocalization. Their results show that while sounds that have an identifiable source can be easily described verbally, this task is difficult for more abstract sounds, especially for people without expertise in sound design. On the contrary, vocal imitations consistently allows to identify the original sound, whatever its level of abstraction.

While these contributions give promising perspectives on the use of vocalizations in sound design, they do not address how the vocalizations interact with gestures. Symmetrically to the problem of retrieving sound from vocal imitations, we now consider the field of Sonic Interaction Design (SID) that focuses on generating new ideas of sounds from vocalizations.

VOCAL SKETCHING In design, *sketching* is a fundamental component of the creative process, as it encourages both thinking and communication in the early stages of the design process (Buxton, 2010). Several authors recently proposed *Vocal Sketching* as the auditory analog to sketching in visual design.

Bencina et al. (2008) introduced the notion of vocal prototyping for generating ideas of gesture-sound relationship for musical performance. Extending this concept to group interaction, Ekman and Rinott (2010) propose *Vocal Sketching* for Sonic Interaction Design (SID) — the domain of design interested in augmenting physical objects and interactions with sound feedback. They conducted a qualitative evaluation through a workshop with designers, which shows that the method allows for the generation of multiple scenarios, often fostering group collaboration. However, the authors stress that the possibilities offered by the voice can both limit the design (when sound cannot be imitated) or bring original ideas (when participants are able to produce specific sounds). Importantly, Ekman and Rinott underline that vocalizing while moving can be socially embarrassing, even if group interaction tends to foster the creative aspects. In a similar way, Tahiroglu and Ahmaniemi (2010) applied vocal sketching to generate sonification ideas for a mobile device.

While this latter work used motion sensors to capture the participants' interaction, they do not use vocalizations for designing the interactive sonification itself. Rocchesso et al. (2015) propose to use gestures in conjunction with vocalization as a practical tool for sketching audio technologies. The Skat-VG project[2] aims to "bridge the gap between utterances and sound models", through the development of models that link gestures and vocal imitations to sound synthesis engines.

VOCAL INPUT IN DMIS While the New Interfaces for Musical Expression community has long been focusing on gestural systems for music performance, several recent work introduced the use of voice as a control strategy in Digital Musical Instruments (DMIs). Although voice and singing has a long history in Digital Musical Instruments (DMIs) (see for example Laeticia Sonami's performances with the *Lady's Glove*), we focus on voice used as an input modality for sound control. Janer (2009)

2 http://skatvg.iuav.it/

proposes singing-driven interfaces for controlling a sound synthesizer, that use syllable segmentation for triggering notes, and associate some characteristics of the vocal sound with the timbre, however without implementing continuous control. In this thesis, we focus mostly on timbre rather than pitched sounds, so we do not give an extensive review of technologies focusing on singing voice analysis and synthesis.

Stowell (2010) proposes two approaches to music making through vocal timbre analysis: a discrete paradigm that remaps beatboxing to a drum synthesizer, and a continuous timbre remapping method based on regression trees that he applied to audio mosaicing using concatenative synthesis. Fasciani and Wyse (2012) extended this continuous paradigm for the control of arbitrary synthesizer. Fasciani and Wyse exploit unsupervised learning to compute perceptual descriptor spaces on both the input vocal sounds and the output of a synthesizer. At runtime, vocal features are remapped onto the sound features, and the associated synthesizer parameters are retrieved by querying the associative database.

Although such methods investigate voice as an input device for musical expression, they do not investigate vocalization in conjunction with movement.

9.1.3 Discussion

From everyday communication to expert methodologies for design and performance, vocalization constitutes an expressive modality of interaction. Although vocalization is widely used as a support for movement practice, pedagogy and performance, few studies have addressed its role in depth, either qualitatively, or quantitatively. While Vass-Rhee underlines a strong collaboration between the vocalizing performers and sound engineers, to our knowledge there exist no computational tool exploiting the voice to support movement pedagogy. In a similar way, while vocalization is consistently used for describing and sketching sound in everyday life, few tools explicitly integrate it as a primary modality for sound design and performance. In this chapter, we consider two applications of vocalization for interaction design, respectively in movement practice and sonic interaction design.

9.2 System Overview

We propose an application of Mapping-by-Demonstration (MbD) where the voice is used in conjunction with body movement to design motion-sound relationships.

One difficulty of the Mapping-by-Demonstration (MbD) approach is the initial design of the sound examples. For example, using a parametric synthesizer requires creating the examples manually. Using recorded sounds simplifies the design process but can make challenging for users to perform a high quality and synchronous gesture while listening to a sound. Voice represent a promising alternative that let users express a variety of

sound variations and that is easy to produce while moving. In movement pedagogy and practice, voice is even used as a way to support movement expression.

In this section, we present a generic system that allows users to craft the relationships between movement and sound using their voice. We derive two applications, detailed in Sections 9.3 and 9.4: a sonic game that plays with vocal imitations, and continuous movement sonification for dance pedagogy.

9.2.1 Technical Description

The architecture of the application, outlined in Figure 9.1, is similar to the approach presented in Section 8.2 for recorded sounds. In this case, the system learns the relationship between gestures and recorded vocalizations, from demonstrations created by vocalizing while moving.

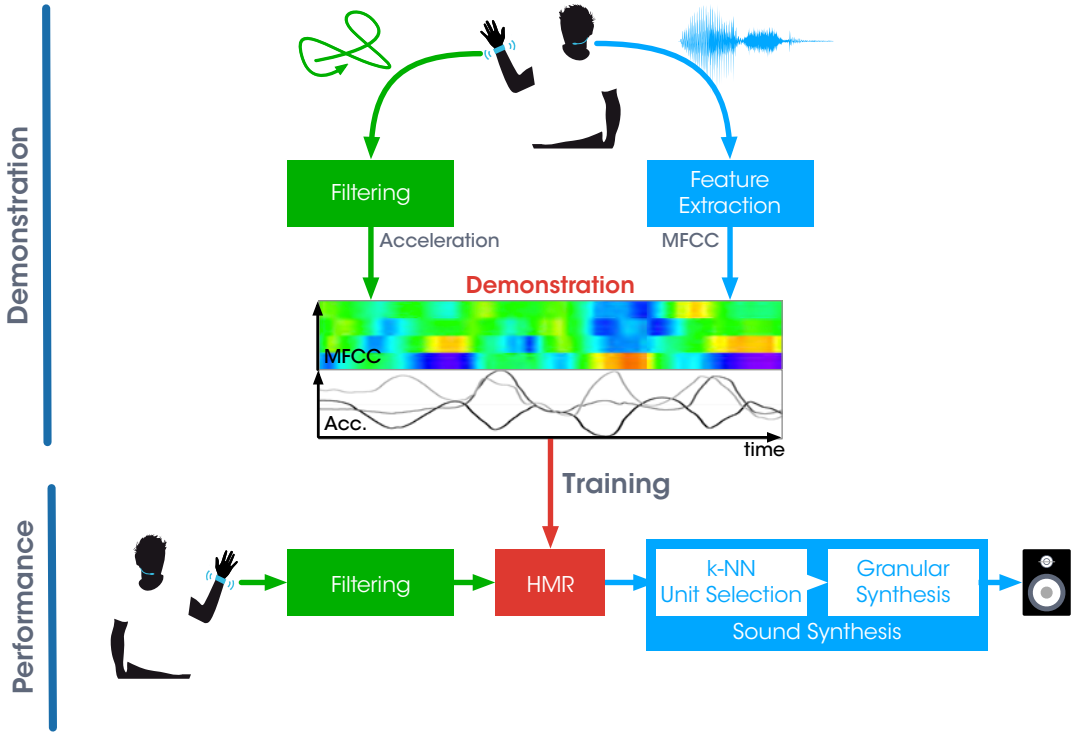


Figure 9.1: Interactive Vocal Sketching

Users build the demonstrations by producing a vocalization while performing a gesture. The system can be used with various types of input devices. Nonetheless, For temporal consistency, it is important to ensure that motion and sound features are regularly sampled, and that both modality are synchronized and have the same sample-rate.

In the following, we investigate body-worn sensors (accelerometers and gyroscopes) and biosensing (in particular, Electromyography (EMG)). Inertial sensors are compact and embeddable, which permits a wide range of uses: they can be directly held in the hand, fixed on body parts, or embedded into objects. In the case of accelerometers, we directly use the raw ac-

celeration — smoothed using a moving average filter, — that encode both dynamic movements and orientations (through the influence of gravity).

The Mapping-by-Demonstration approach requires an analysis/synthesis framework for the voice that shares a common parametric representation. As a first step in the system design, we use descriptor driven sound synthesis in conjunction with a description of voice using sound descriptors. We consider Mel-Frequency Cepstral Coefficients (MFCCs), that describe the shape of the spectrum on a fixed-size window of audio data. Mel-Frequency Cepstral Coefficients (MFCCs) are widely used in Speech recognition and synthesis for their ability to describe the timbre of the voice (See Section 2.5).

Movement and sound Features are recorded synchronously and equally sampled to form the multimodal demonstration. We train a Hidden Markov Regression (HMR) model on the multimodal sequence of frames composed by the concatenation of motion features and sound descriptors. The model encodes the relationship between motion and sound parameters, taking into account their common temporal structure. The model is trained with the Baum-Welch algorithm that maximizes the likelihood of the joint parameter sequences. Once trained, the model can be used for simultaneous gesture recognition and sound parameter generation. We convert the joint probability density function (pdf) of each hidden state to a conditional distribution expressing the distribution of sound parameters as a regression over motion parameters. In *Performance*, we use Hidden Markov Regression (HMR) to generate the sound parameters associated to an input movement, using the method described in Section 6.2.

At each time step, we jointly estimate the likelihood of the gesture given the trained model and generate the sound parameters. The sound parameters are then continuously streamed to the synthesis engine.

9.2.2 Voice Synthesis

Most HMM-based parametric speech synthesis systems use a source-filter model of voice production. The source excitation switches between harmonic and noise generators (for voiced/unvoiced sounds), that are filtered by a spectral model based on Cepstral Coefficients (See the STRAIGHT method (Kawahara et al., 1999)).

Our goal in gesture-based interaction with vocalizations differs from speech synthesis. Rather than having a generic voice synthesizer, we aim to develop models that can be quickly learned, adapted and personalized from few training examples — possibly a single demonstration. Moreover, our goal is oriented towards expressivity rather than intelligibility and realism.

As a first iteration in the development of the system, we propose a sample-based sound analysis-synthesis approach drawing upon descriptor-driven corpus-based sound synthesis. We build a corpus of vocal sounds from the demonstrations, where each recording is associated to its sequence of MFCCs. In performance, HMR continuously streams the generated sound descriptors to a descriptor-driven synthesis engine.

Similarly to CataRT, we use a *k*-Nearest-Neighbor (k-NN) search within the entire corpus to select the audio frame that matches the target vector of sound descriptors. We use the indices of the selected frames to drive a granular synthesis engine that continuously synthesizes the vocal sounds contained in the buffer. Several parameters of the grains can be tuned to adjust the quality of the sound synthesis. Envelope parameters such as duration, attack and release time, are combined with the period of the granular synthesis specify the density of grains, allowing to implement various sound 'qualities'.

A video demonstrating the system for gesture-based control of vocalizations can be found online.[3]The system is also currently used by Greg Beller in his musical research residency at Ircam. The "Synekine Project" (Beller, 2014) explores voice and movements in contemporary performance using interactive environments. Several Examples of work-in-progress are available on Greg Beller's website, notably "Wired Gestures"[4]that uses the system based on HMR.

9.3 The Imitation Game

We presented a generic system for designing sonic interactions based on vocalizations and gestures. We now present a playful application of the system to an 'imitation game' that we presented at the SIGRRAPH'14 Conference held in Vancouver, Canada. The proposal, called "MaD: Mapping by Demonstration for Continuous Sonification", is outlined in Françoise et al. (2014b) and will be featured in the 'Demo Hour' section of the March-April 2015 issue of ACM Interactions.[5]

Our proposal for SIGGRAPH was composed of two applications of Mapping-by-Demonstration to interactive installations targeting a broad audience. Both setups were jointly presented in the *Studio* and *Emerging Technologies* spaces. We reported on the first installation, presented at *Studio*, in Section 8.3: it was the proposition of a system for intuitive hand gesture interaction with environmental sounds. We now detail the second installation, presented at *Emerging Technologies*, that implemented a game based on vocal and gestural imitations.

9.3.1 Context and Motivation

We started investigating vocalization as a use-case of the Mapping-by-Demonstration (MbD) approach following several experiments with physical modeling sound synthesis and recorded sounds. In both cases, the process of designing sound and executing gestures while listening to sound poses difficulties in producing high-quality, consistent and synchronized

3 http://vimeo.com/julesfrancoise/mad
4 http://www.gregbeller.com/2014/02/wired-gestures/
5 http://interactions.acm.org/

demonstrations, even when gestures are performed while listening. On the contrary, co-producing vocal sounds and gestures is ubiquitous in everyday life, such as gesturing to support verbal communication or combining gestures and non-verbal vocal sounds to produce imitations. Voice therefore represent an efficient way to provide MbD systems with consistent demonstrations of the relationships between movement and sound. Moreover, as outlined in Section 9.1.2, vocalization becomes a primary modality of describing and interacting with sound synthesis. In particular, the Skat-VG European project aims to study how to sketch audio technologies from joint vocalizations and gestures (Rocchesso et al., 2015).

During the development and experimentation with the interactive vocal sketching system, we made several observations. First, we noticed that the sound feedback helped to repeat specific gestures very consistently, as soon as the user was able to reproduce the vocal sound. Second, we observed that imitating another person's gestures with accuracy was not trivial. In particular, using inertial sensors for motion capture implies that the movement is described in terms of dynamics rather than spatial trajectory, and such dynamics are particularly difficult to imitate from visual information only.

To fit the context of a public installation, we derived an entertaining application based on vocal imitations. We proposed to 'gamify' the application in order to enhance participants engagement and limit the possible awkwardness of producing vocal sounds in public. In the subsequent sections, we describe the installation and the structure of the game, and report qualitative results from the feedback gathered during the presentation at SIGGRAPH.

9.3.2 Overview of the Installation

The installation was composed of two parts: a 'demo' mode allowing to quickly introduce the system and give users the opportunity to interact. Then, users could involve by teams of two in a 'game' mode which goal was to reproduce the vocalizations accurately under timing constraints.

DEMO MODE The *imitation game* is a two-player game that engages users in imitating each other's gestures to reproduce vocal imitations. The setup of the game is illustrated in Figure 9.2. The two players are facing each other on both side of the interface. We used the interface to display a simple visual feedback of the motion and the microphone sound. The red and green buttons were used for triggering the recording of the demonstration and the performance mode, respectively.

The interaction flow of the installation, as described in Figure 9.3, starts with a participant jointly producing a gesture and a vocal imitation. Once the demonstration recorded, we invite the first player to reproduce the gesture to attempt to resynthesize his/her vocalization, and we subsequently ask the second player to produce an imitation from the observation of the first player. We used this 'demo' mode to introduce the system and give a first overview of its possibilities.

(a) (b)

Figure 9.2: Pictures of the setup of the imitation Game at SIGGRAPH'14. The players were facing each other, on each side of the screen. The red and green push buttons were used to trigger the demonstration and performance modes.

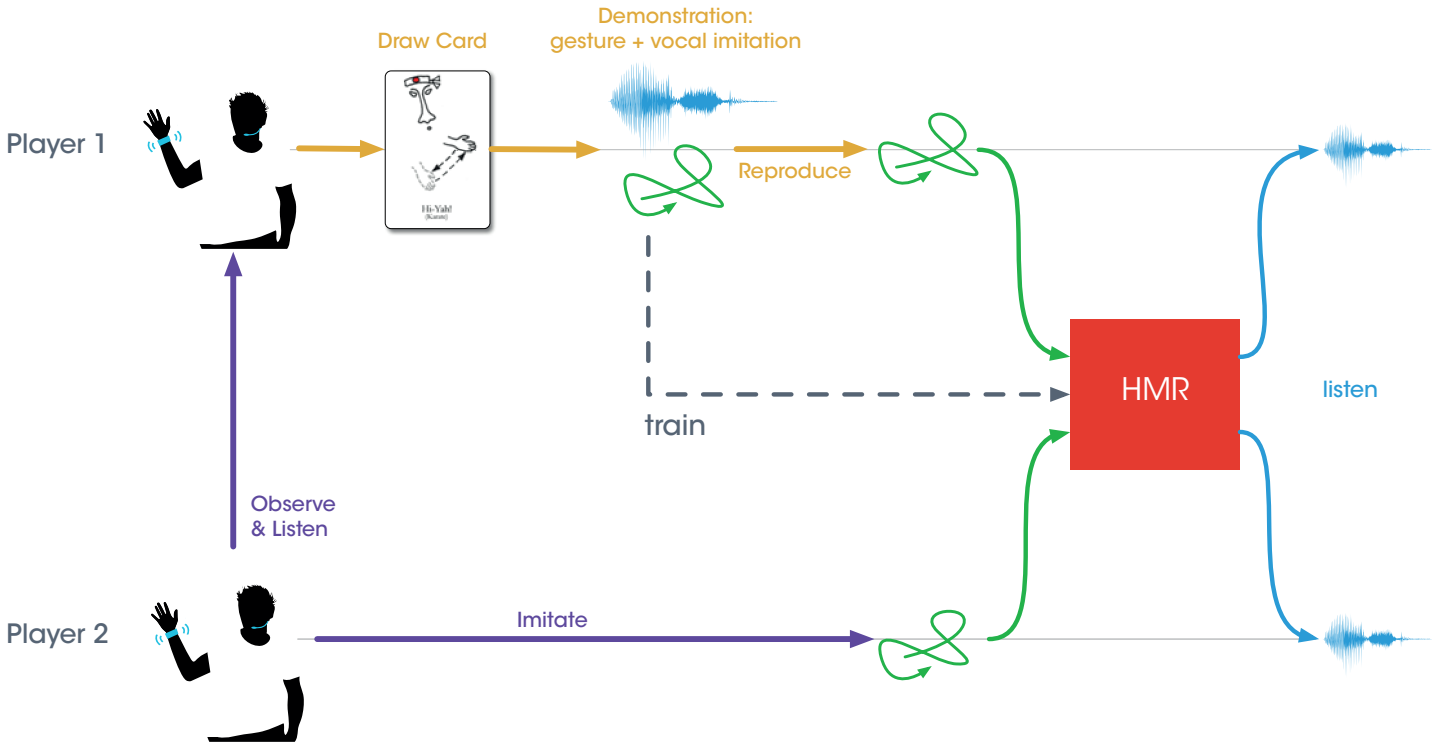


Figure 9.3: Principle of the Imitation Game. One participant draws a card and perfoms a joint gesture-vocal imitation. Once the system trained, he/she reproduces the gesture to reinterpret the sound. The second participant imitates his/her movement, to try to achieve the same vocal imitation.

To give participants some inspiration, we created a set of *action cards* that gave a pictorial representation of an action with its associated sound

as a phonetic indication. The action cards were selected and developed in brainstorming and preliminary tests, and the final set of 17 cards was illustrated by Riccardo Borghesi (see Figure 9.4). Several cards explicitly referred to sound producing human actions (*butcher, drinking, rally, swatter, theremin, uppercut*) while others represented co-produced gestures and vocal sounds (*karate, yawning, yummy, snoring, circle*), or environmental sounds (*fly, lightning, wave, wind, wipers*).

Figure 9.4: The action cards used in the imitation game. Cards were mainly a support to users for creating joint gestural and vocal imitations.

GAME MODE The second mode implements a game where the players' goal is to produce sequences of their imitations as accurately as possible. The game starts with the definition of the mappings: each participant can record two or three gesture-sound imitations from a set of randomly selected cards displayed on the screen. The process is similar to the demo mode, and participants can adjust their imitation and imitate each other before starting the game. This phase provides 4 to 6 imitations that form the basis of the sequence game.

A screenshot of the sequence game is presented in Figure 9.5. When the game starts, we display on each side of the screen the same sequence of cards, drawn randomly without repetition from the recorded imitations. The cards then successively flash on a regular tempo, triggering the appropriate mapping — in order to simplify the game, no recognition is included in the installation. Each time a card flashes, the two participants must reproduce the gesture to generate the sound as accurately as possible. After each set of four cards, new actions are randomly drawn and the tempo accelerates. The game ends when the tempo gets too fast to allow players to reproduce the gestures.

To enhance participants engagement, each player has to maximize a score associated to her performance of the gestures. The score, computed from the recognition parameters, was designed to reflect the player's reproduction accuracy. During the game, we examine the time progression estimated by the HMM, and allocate points when the time progression linearly increases, meaning that the system follows the player's gesture. Play-

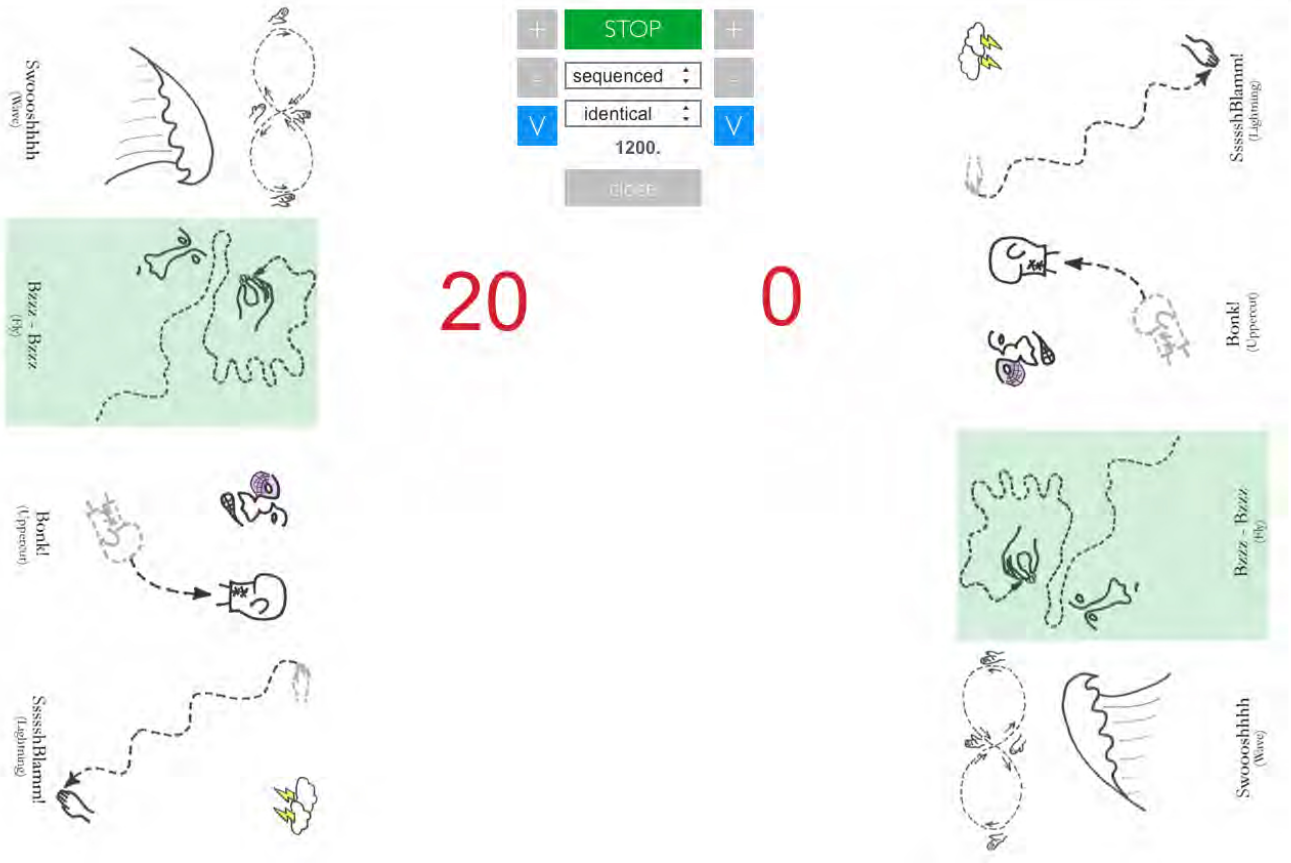


Figure 9.5: Screenshot of the interface of the Imitation Game

ers won additional points when they could reproduce the entire gestures with high accuracy.[6]

9.3.3 Method

DATA COLLECTION We collected data during the five days of presentation at SIGGRAPH'14 *Emerging Technologies*. We first invited attendees to try the system in demo mode to give a brief overview of the system's possibilities and applications. Then, we invited interested participants to perform the full game with two players that each recorded two or three imitations and simultaneously performed the sequence game. In case a single participant was present, the experimenter acted as the second player. At the end of the game, we asked the participants if they agreed to give their motion, vocal, and optionally video data for research purposes. The agreement was directly completed on the interface, and the participants had the possibilities fill in their age range.

For each team of participant, we recorded data in the game mode:

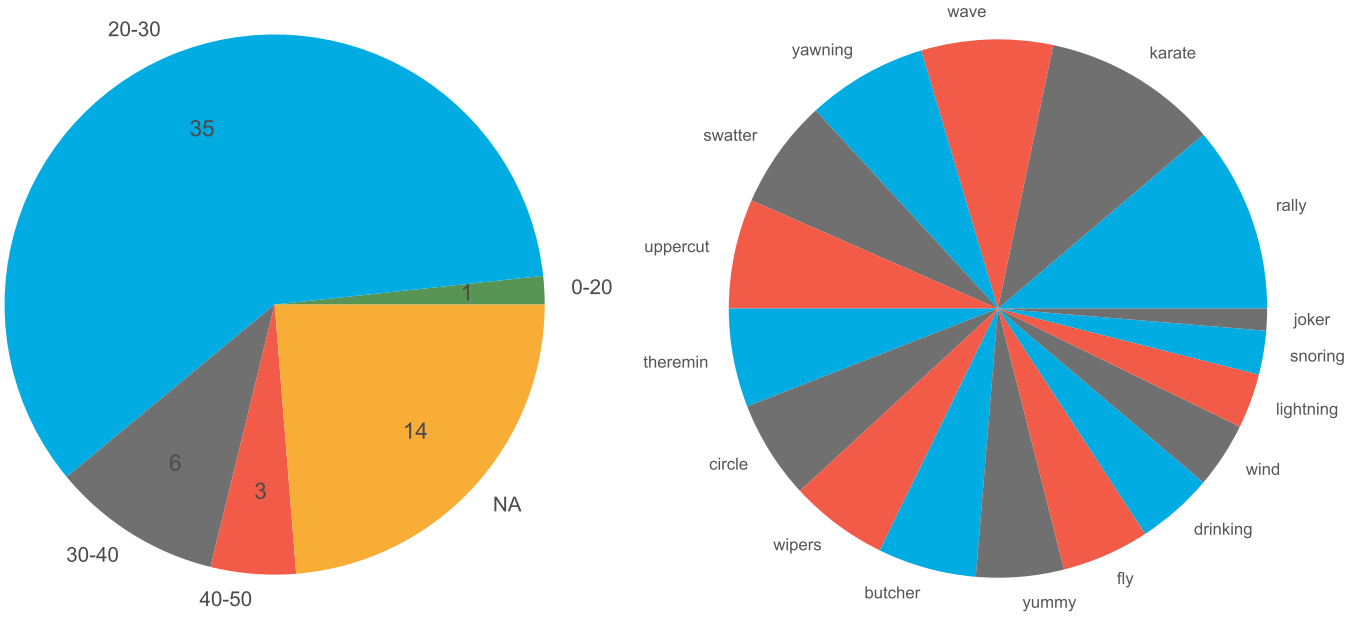
6 We stress that the purpose of the score is to foster the player's engagement, and we do not pretend that it provides an accurate measurement of the gesture imitation accuracy

DEMONSTRATION We recorded the 3D acceleration, the audio of the vocalization, and the associated sequence of MFCC, of each of the 4 to 6 imitations performed by the team of participants.

PERFORMANCE We recorded the 3D acceleration, the sequence of MFCC generated by the HMR model, the synthesized audio, and the sequence of action cards for each game.

We also collected the data recorded in the demo mode: each demonstration and the following performance phases where participants could explore the sound feedback and attempt to reproduce the gesture.

PARTICIPANTS We selected the teams of participants who both agreed to the consent form. We discarded the sessions that presented recording errors, missing action cards (inferior to four cards for a game), or incomplete game sequence. Discarding the experimenters from the final set of participants, we collected data from 59 attendees, whose age repartition is reported in Figure 9.6a. Most participants were aged under 40, and came from the fields of computer graphics, animation, motion picture. Participants could chose among a random set of action cards the imitations they wanted to use in the game. The frequency of use of each action card is presented in Figure 9.6b. All action cards were selected at least 10 times, and two actions (*karate* and *rally*), were used more than 30 times.



(a) Repartition of the age of participants

(b) Frequency of choice of the action cards for the game

Figure 9.6: Information on participants and their choice of action cards

ANALYSIS We propose to investigate participants' consistency in gesture imitation as well as their ability to reproduce the original vocalization. We analyzed the gestures performed by the participants in the sequence game. The game consists in a sequence of 24 *segments* associated with action cards, that the participants must reproduce. The duration of each segment is determined by the timer of the game, which decreases along the game. We derive two approaches for gesture analysis: a basic investigation of descriptive movement statistics, and a continuous measure of based on log-likelihoods.

Descriptive Movement Statistics

First, we compute a set of statistics on simple descriptors of both movements and sounds. This aims to give insights into the type of gestures and vocalizations performed by the participants for each action card. We compare four quantities, computed on the demonstrations. *Duration* relates the total duration of the recording of the imitation. *Maximum Energy* is computed as the maximum of the norm of the gesture's energy[7] along the demonstration gesture. For sound, we compute both the *maximum loudness* along the vocal imitation, and the *Harmonic/noise Ratio* that reports the ratio of energy between the harmonic content and the residual noise — that relates to the voiced or unvoiced qualities of the vocalization.

Log-Likelihoods

Second, we investigate log-likelihood as a continuous measure of the gesture accuracy in reproducing the original demonstration. The procedure for analyzing a team of two players is as follows. We train a single 30-state HMM for each recorded gestural imitation, with a left-right topology and an additional transition from the last state to the first state. Each model is therefore associated to a class (action card), and the class is know for each segment in the sequence. For each participant, each segment, we compute the cumulative log-likelihood of the appropriate class over the entire segment (normalized by the segment length). The likelihood gives a continuous measure of the accuracy of the imitation of the gesture with respect to the demonstration. We perform the same analysis on the sequences of sound descriptors, to evaluate the similarity of the generated sequences of MFCCs to the demonstration.

9.3.4 Findings

DESCRIPTIVE MOVEMENT STATISTICS First, we report descriptive statistics computed on the demonstrations for each action card. Both the duration of the imitation and the peak of energy of the gesture, reported in Figure 9.7, give qualitative insight on how the gestures are performed for each class. It allows to identify several types of gestures: *butcher*, *karate*, *swatter*, and *uppercut* can be considered impulsive: they have a high energy a short duration. On the contrary, environmental sounds tend to be longer with low energy, as the gestures mimic the evolution of the sound. Observing the Harmonic/Noise ratio of energy,

7 Our estimation of the movement's energy is based on the norm of the derivate of the filtered acceleration signals.

in the top right plot of Figure 9.7, we note that several action cards are more likely to be performed with voiced sounds: *circle*, *rally*, *theremin*, *wind*, *wipers*, *yawning* and *yummy*. More qualitatively, we observed a large variety of strategies during the presentation at the conference, from the perspectives of both gestural and vocal imitation.

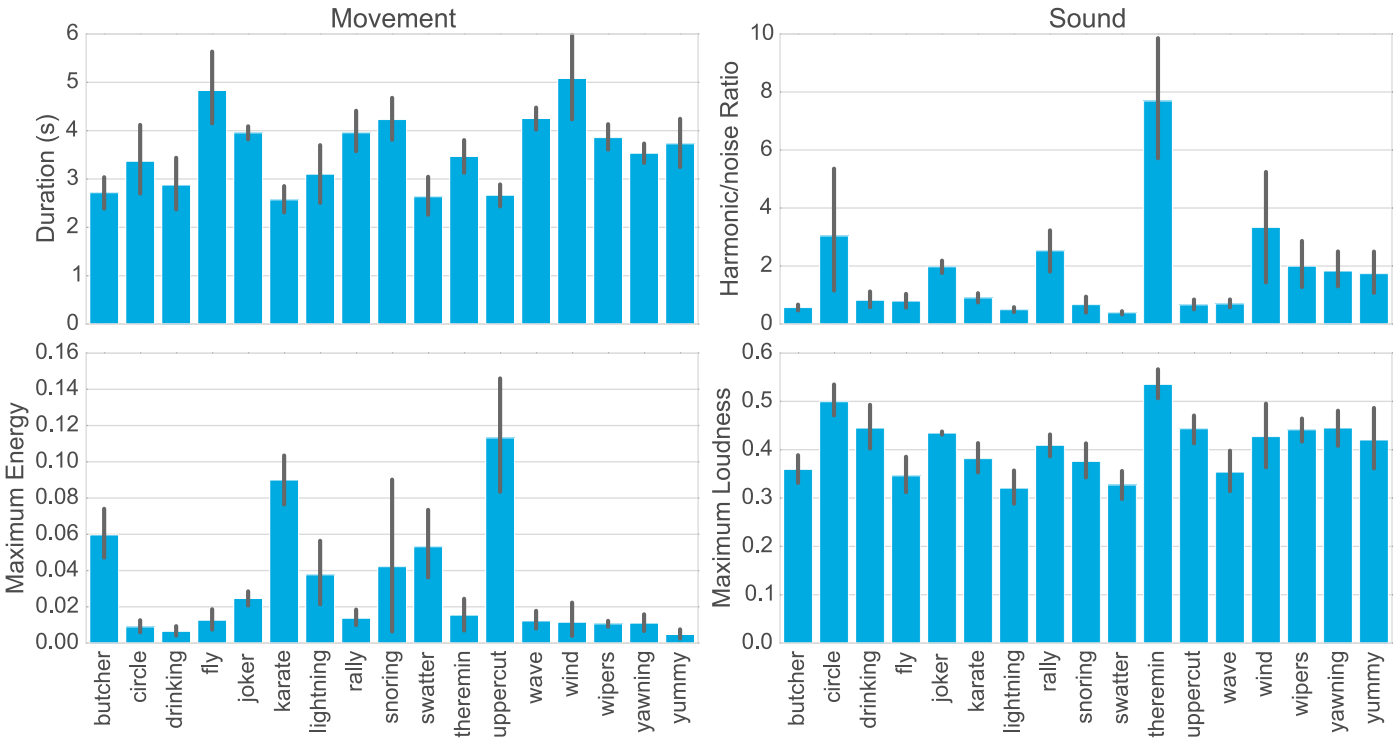


Figure 9.7: Movement duration and energy for each action card

ILLUSTRATIVE EXAMPLE We report in Figure 9.8 an example of game data. The figure depicts in the two first rows plots the acceleration and audio recordings of the 6 imitations used in the game. We can observe that depending on the action card, certain gestures and vocalizations are impulsive (*butcher*, *swatter*), while other present a clear periodicity (*joker*, *wipers*).

The two bottom plots respectively report the acceleration signal and synthesized audio from the sequence game. While we can identify some patterns in the sequence game that reproduce the initial gestures, we can observe that the motifs are often repeated several times in each segment. The audio waveform highlights that the synthesized sound is continuously controlled by the gesture, and is modulated in amplitude by the energy of the movement. As the duration allocated for playing each card decreases, we observe that players tend to repeat the gestures more often and faster, which leads to a global increase of the motion energy.

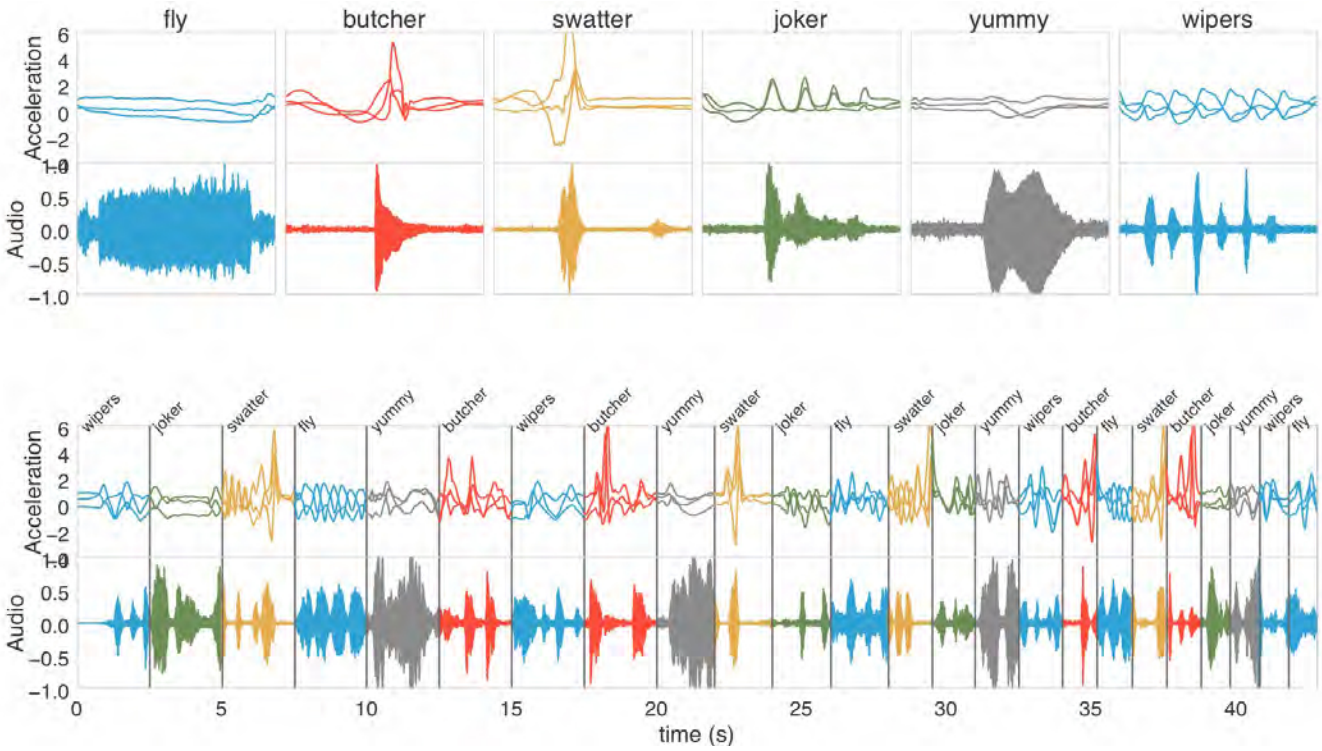


Figure 9.8: Example of Game data. The set of imitation is composed by 6 gestures (top) and their associated vocalizations (bottom). The middle plots represent the acceleration and synthesized audio during the game.

LIKELIHOOD AS A MEASURE OF ACCURACY

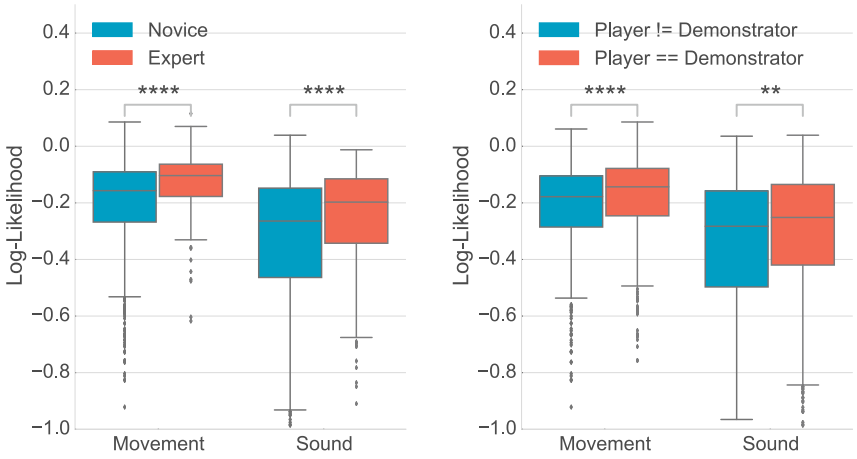
We compare the cumulative log-likelihoods computed on each segment during the sequence game, for both a HMM trained on the acceleration, and a HMM trained on the generated MFCCs. In the following, we use nonparametric statistical tests for null hypothesis testing, considering that log-likelihoods are very unlikely to be normally distributed.

Expertise

First, we consider the difference between novice users (the conference attendees), and expert users (the experimenters). The log-likelihoods obtained with movement and sound across action cards and participants is represented in Figure 9.9a. We identify a significant difference between the two groups of users: experts yield a higher log-likelihood than novice users. We ran a Mann-Whitney's U test to evaluate the difference in log-likelihood. We found a significant effect of expertise, both on movement features ($U = 135477$, $Z = -7.11$, $p < 0.001$, $r = 0.17$), and on sound features ($U = 154281$, $Z = -4.51$, $p < 0.001$, $r = 0.11$). In the following, we discard all expert participants and perform the analysis on novice users only.

Demonstrator vs Player

Figure 9.9b reports the log-likelihoods across action cards and participants according to the criterion that the player is the same as the demonstrator who created the initial imitation. Similarly, we observe that the log-likelihoods are lower when the player is not the one who recorded the initial demonstration. We found a significant effect of the match between the



(a) Log-likelihood according to expertise

(b) Log-likelihood according to the match between the player and the demonstrator for novice Participants

Figure 9.9: Log-likelihood according to the player's expertise and match with the demonstrator. The black line represents the 95% CI around the mean.

player and the demonstrator, both on movement features (U = 214161, Z = -4.73, $p < 0.001$, r = 0.13), and on sound features (U = 228435, Z = -2.88, $p < 0.01$, r = 0.08). These results suggest that participants reproduce the gesture imitations more accurately when they performed the initial demonstration. It is interesting to note that this effect is also guaranteed for the generated sound parameters, though with a lesser effect size.

Action Cards

As depicted in Figure 9.10, the Log-likelihood greatly varies among action cards. Moreover, the difference between the groups varies with the imitated actions. While several action cards show a significant difference between the groups whose player is demonstrator and whose player is different, several gestures are equally performed by both participants. Some gestures present a very important difference, for example *butcher*, *lightning*, *rally* or *wave*, and might indicate very idiosyncratic ways of performing certain actions, with respect to the motion capture system.

> The Log-likelihoods computed on the segments of the sequence game shows a significant difference between players that did record the demonstration and the other players, which indicates that demonstrators are able to reproduce their gestures more accurately. This difference varies with the action cards, suggesting that all gestures might not have the same reproducibility or the same level of idiosyncrasy.

TIMING CONSTRAINTS The duration allocated for performing each action card decreases along the game. The first 8 segments last 2.5 seconds, and the timing then decreases every 4 cards (2 s, 1.5 s, 1.2 s, 1 s). We now investigate how users' gestures evolve with the timing constraints along the game.

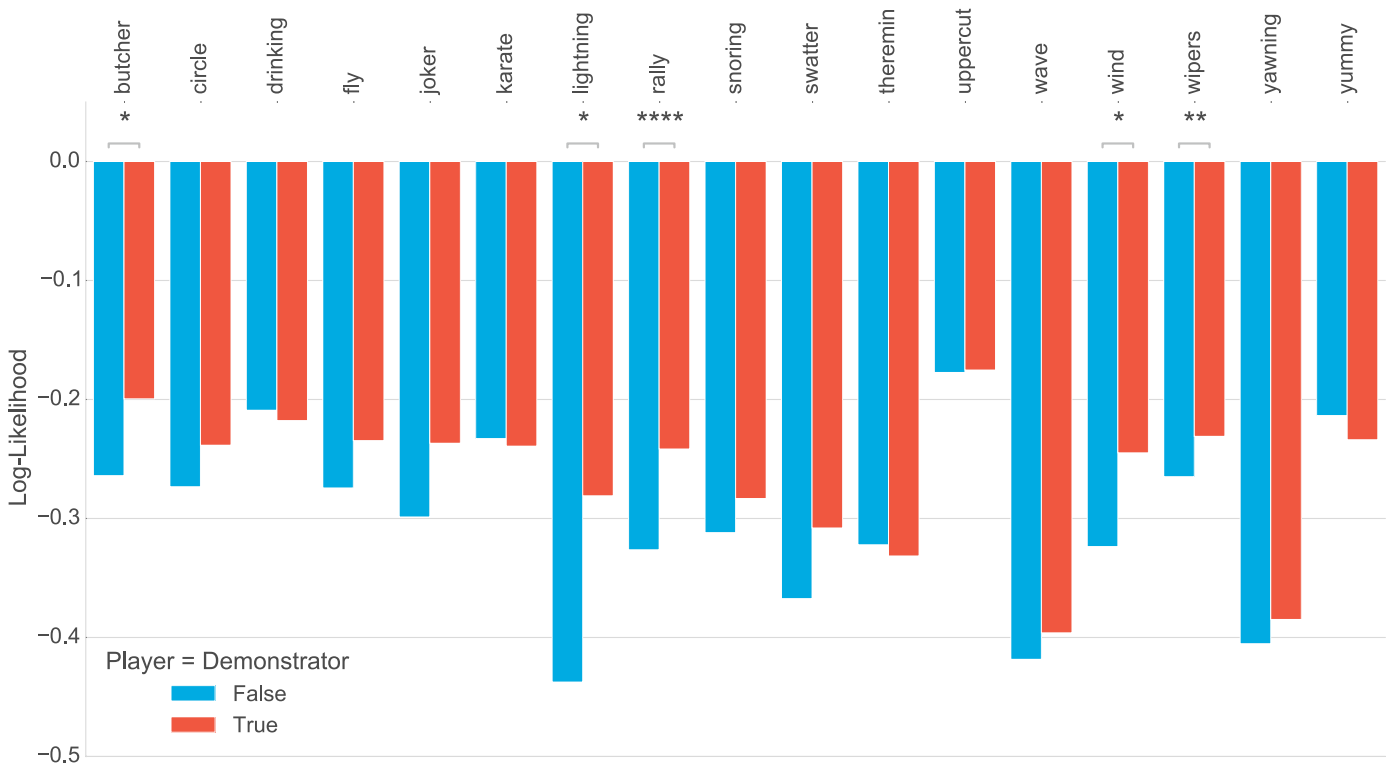


Figure 9.10: Log-Likelihood by action card

Figure 9.11 reports the log-likelihood according to the duration allocated for performing the gesture, across all participants and action cards. We observe a global descrease of the log-likelihood as the duration alllocated for reproducing the gesture decreases. The result a post-hoc analysis using

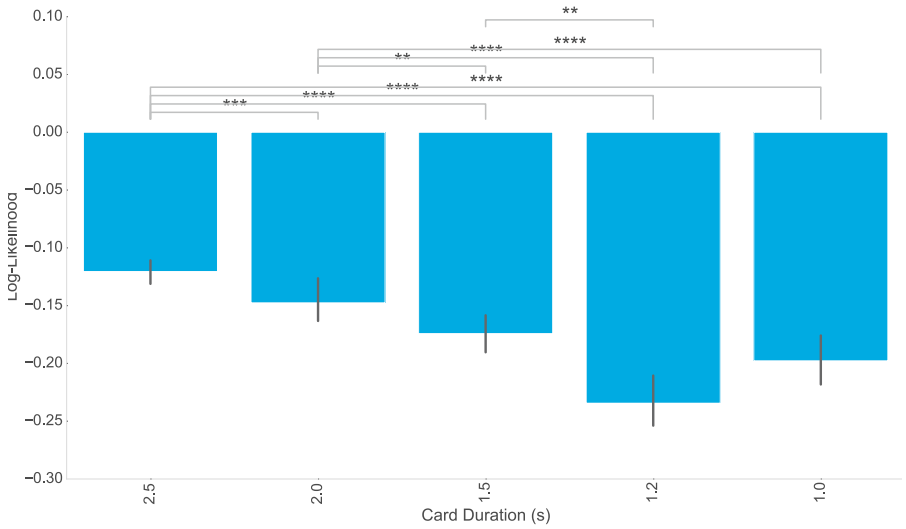


Figure 9.11: Log-likelihood by card duration for Movement and Sound. Starts indicate statistically significant relationships, according to independent t-tests.

paired Mann-Whitney U tests with Bonferroni correction indicate that the log-likelihood significantly decrease at each reduction of the allocated duration, under $p < 0.05$, except for the last segment. The results indicate that the performance degrades as the duration allowed to perform the gesture decreases.

If we consider the evolution of the energy of the movement during the game, we observe that energy increases as the segment duration decreases, from 2.5 seconds to 1.2 seconds. These results correlate with the observation of the players, that tend to repeat the gesture several times in each segment to accumulate points. As the duration decrease, the tension of the players increase and lead them to move faster, resulting in high acceleration values. The decrease of the energy in the last blocks is due to the players giving up moving when the duration of the card is too short to adapt their gestures. The decrease of the energy in the last blocks is due to the players giving up moving when the duration of the card is too short to adapt their gestures. We can correlate these variations of energy to the decrease in log-likelihood along the game, that indicate the inability of the participants to reproduce the gesture dynamics when they increase the intensity of their gesture.

> The log-likelihood significantly decreases as the duration allowed for reproducing the gesture decreases, which indicates that players become less accurate at imitating the initial demonstration when the game accelerates. This decrease can be correlated with the acceleration of the game timing that leads the players to move with more energy, and therefore can't reproduce the motion dynamics.

QUALITATIVE OBSERVATIONS: GESTURE ADAPTATION WITH SOUND FEEDBACK

We now qualitatively discuss several observations we made during the presentation of the installation. We observed a very wide variety of strategies for reproducing both gestures and sound imitations. While some participants attempted to produce an imitation of the sound associated with the action card, several participants used verbal sound or onomatopoeia, directly uttering the phonetic indication of the action card. For most of the participants, jointly producing the gesture and the sound was found easy and intuitive. Many participants felt comfortable with using gestures and vocal imitations, and reported that the game was entertaining. The set of action cards encouraged participants to exaggerate and caricature the action-sound relationships, that added to the fun of the installation. Several attendees, however, were not comfortable with using the voice in public, especially when combined with an involvement of the body.

Most importantly, we observed that the interactive system provides a rich feedback for adapting and learning gestures. Often, participants manage to reproduce the dynamics of the demonstrator's gesture by iteratively exploring the movement and its relationship to the sound feedback. We found that combining the auditory feedback with verbal guidance allowed to quickly converge to the correct motion dynamics.

To illustrate this process, Figure 9.12 reports the vocal imitation of a player, and the adaptation of the second player using the sound feedback

in 'demo' mode.[8] The figure depicts two attemps to reproduce the vocalization by imitating the gesture of the first player (shaded segments in Figure 9.12). For the first trial, we observe that the acceleration pattern presents the same global trajectory as the demonstration, but with different dynamics. The generated sound is very different from the demonstration, and we can observe that the model is not able to follow the gesture smoothly (see the time progression, bottom plot). The second trial reproduces more accurately the dynamics of the acceleration pattern. In this case, the player managed to reproduce the gesture accurately. The time progression evolves smoothly along the model, and the generated sound is very similar to the original demonstration — except for the additional pitch shift effect. In many cases, we observe a quick adaptation of the participants along the trials. Often, the players were able to reproduce the target sound in a few attempts.

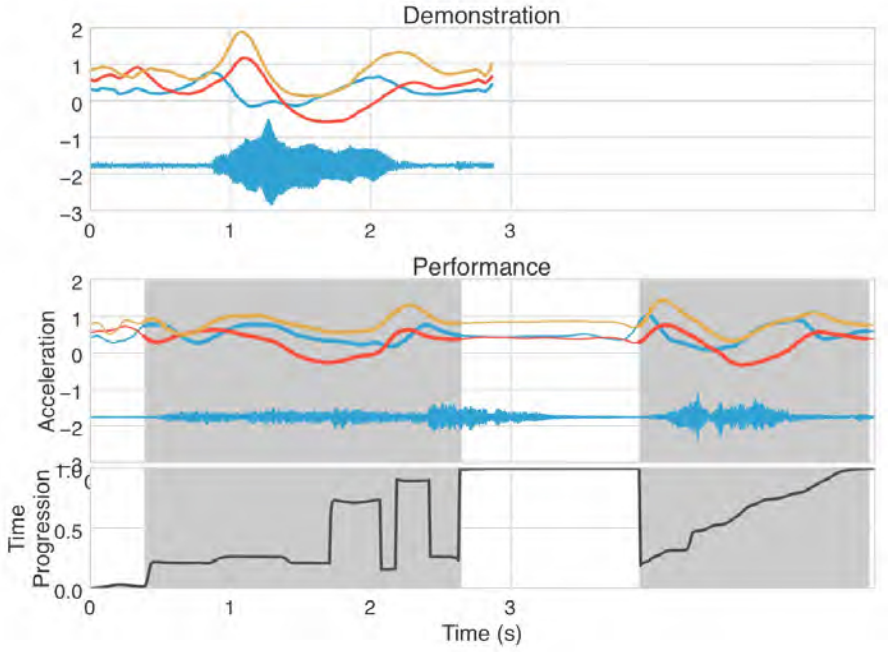


Figure 9.12: Example of acceleration and recognition data in demo mode. The top plot represents the demonstration, composed of acceleration and audio signals. the middle plot depicts the recording of the participant attempting to reproduce the initial gestures, and the bottom plot reports the time progression estimated with a HMM. Gray areas highlight the participants attempts to imitate the gesture. The audio recordings of the demonstration and performance are available online: http://julesfrancoise.com/phdthesis/#vocalgame_example

8 The audio recordings of the demonstration and performance are available online: http://julesfrancoise.com/phdthesis/#vocalgame_example

9.3.5 Discussion and Future Work

We reported the design and evaluation of a system implementing a Mapping-by-Demonstration approach for a public interactive installation. The system draws upon joint performances of gestures and vocal imitations to design gesture-based sound control strategies. The installation, presented at SIGGRAPH'14 *Emerging Technologies*, featured an imitation game where two players had to imitate each other's gestures to attempt to resynthesize their vocal imitations.

Expertise and timing constraints are crucial factors in users' ability to reproduce gestures. Qualitative observations support the idea that interactive sound feedback helps reproducing gestures dynamics that are not easily apprehended using the visual modality only. We would need to conduct a more controlled experiment to evaluate the effect of the sound feedback to reproduce specific motion dynamics.

9.4 Vocalizing Dance Movement for Interactive Sonification of Laban Effort Factors

We now address the second facet of vocalization we highlighted in Section 9.1: the use of vocalization to support movement practice. This chapter reports the results of an exploratory workshop with dancers that applies the generic system we detailed in Section 9.2 to the continuous sonification of movement qualities. The system uses HMR to map between movement and vocalizations, where the demonstrations are created by expert performances of Laban Effort Factors. We report on an exploratory study where we use the system in a teaching session with dancers.

This section is an adaptation of a previous publication: "Vocalizing Dance Movement for Interactive Sonification of Laban Effort Factors" (Françoise et al., 2014c), presented at the *ACM DIS'14* Conference, the International Conference on Designing Interactive system, held in Vancouver, Canada, June 21–25, 2014.

The work that we report here is part of an ambitious research agenda focusing on movement expressivity in HCI through the use of movement qualities as an interaction modality. Fdili Alaoui et al. (2012) describe movement qualities as a qualitative attribute of moment produced by dynamics and defining the ways movement is executed. In this paper, we report on a specific aspect of this research: investigating the use of interactive sound feedback to reflect and guide the performance of movement qualities as defined in the Effort category of the Laban Movement Analysis (LMA) framework. LMA formalizes movement qualities as Effort, a category that describes how a movement is performed with respect to the mover's inner attitude or intention.

We propose a methodology for the interactive sonification of Effort Factors that draws from vocalizations performed by Certified Laban Movement Analysts (CMAs). Our interactive system is built upon machine

learning methods that learn the mapping between movement and sound from expert performances. We evaluated our movement–sound interaction models and the associated sonification systems in a workshop where dancers were taught to perform and experience Laban's Effort Factors. The workshop used bodystorming and focus group open-ended interviewing techniques to elicit participant feedback regarding design, technological, and experiential issues of voice-based sonic interactions in dance pedagogy. This format allowed us to assess the participants' experience of the interactive sonic feedback and to establish guidelines for further development of sonic systems dedicated to movement qualities.

9.4.1 Related Work on Movement Qualities for Interaction

Few interactive systems exploit movement qualities as an interaction modality, particularly for dance or artistic installations, and they rarely include interactive sonic feedback to support movement expression. Camurri et al. (2000b) developed a framework for music and dance applications to analyze and classify expressive gestures along Laban Effort Factors using the Eyesweb platform. (Schiphorst, 2009) used Laban's Effort Factors to enhance body awareness and the æsthetic experience of interacting with tangible digital media. Fdili Alaoui et al. (2012) have recently shown that movement qualities can enhance user experience and exploration in interactive installations, and such a system was evaluated and used in dance pedagogy (Fdili Alaoui et al., 2013). Maranan et al. (2014) modeled Efforts using a single-accelerometer system, for interactive visualizations in dance performances and installation. Mentis and Johansson (2013) proposed a study that aims to situate the perception of Effort Factors, through a Kinect-based system for an improvisational dance installation in which the recognition of audience members' Effort Factors trigger musical events. However, to the best of our knowledge, no system addresses the sonification of Laban's Effort Factors for dance pedagogical purposes.

9.4.2 Effort in Laban Movement Analysis

LMA is a system that provides rigorous models for the description of movement, its function and its expression through four components, defined as Body, Effort, Space and Shape. In this paper, we investigate the qualitative aspects of movement that conveys movement expressiveness, as defined in the Effort category of LMA. Effort can be experienced and observed as an attitude shift that reveals the mover's intent in response to the environment (Laban and Lawrence, 1947). It encompasses 4 discrete Factors: Space, Time, Weight, and Flow. Each Effort Factor is thought of as a continuum between two opposite 'Factors' in which movement can vary and thus reveal different 'qualities' (Laban and Lawrence, 1947). Space is related to one's attention to the surrounding environment either through scanning the whole environment (*Indirect Space*) or focusing on a single element (*Direct Space*). Time is related to one's inner attitude to time expressed either through acceleration (*Quick Time*) or deceleration (*Sus-*

tained Time). Weight is related to one's resistance to gravity through increasing (*Strong Weight*) or decreasing pressure (*Light Weight*) using the muscular tension. Flow is the experience and expression of the ongoingness of movement that determines how movement is released (*Free Flow*) or controlled (*Bound Flow*).

BREATH SUPPORT IN LMA Breath is the first form of human movement, and breath continues to support the development and expression of all human movement throughout our life cycle. According to Irmagard Bartenieff, one of the major historical figures of LMA, "Movement rides on the flow of Breath" (Bartenieff, 1980). The use of Breath in LMA allows the human body to access a large palette of movements, by supporting the phrasing of movement and the full body shaping. Bartenieff emphasizes the crucial role of vocalization as an amplification of Breath in achieving a fluidity and expressivity in movement.

9.4.3 Movement Sonification based on Vocalization

We recorded two CMAs, *Karen Studd* and *Philippe Seth*, vocalizing and performing movements with Sustained or Quick Time Effort or Strong or Light Weight Effort. Each CMA was asked to vocalize throughout her movement performance using breath support with the intent of producing a sound quality that subjectively 'represented' the performed Effort. Each Vocalization was recorded synchronously with multimodal sensor data. Previous experiments allowed us to derive a set of sensors and movement features that are useful to characterize Effort Factors through the formalization of expert CMAs' observational process. In this study we chose to focus on Time and Weight Effort Factors, because they are the most accurately recognized in real-time using the proposed sensors and features.

We used two different types of sensors to record the movement data: a 3D accelerometer attached to the right wrist and an electromyography sensor (EMG) attached to the forearm (see Figure 9.13). Data was sampled at 200Hz and transmitted wirelessly to a computer. A microphone, connected to the same computer, was used to record the CMA's vocalization. The movement features selected to represent Time and Weight Effort Factor are respectively:

Time Effort Factor: magnitude of the derivative of the acceleration measured with a 3D accelerometer placed on the right wrist. This feature correlates with the sense of acceleration and deceleration that CMAs use to observe Quick versus Sustained Time Effort.

Weight Effort Factor: intensity of the muscular activation evaluated using a non-linear Bayesian filtering of Electromyography (EMG), captured with an EMG sensor placed on the right forearm. The muscular tension variation correlate to the experience of Strong versus Light Weight Effort.

We developed two separate models for Time or Weight Effort Factor. Each model was trained using 24 pre-recorded vocalizations and performance of movements with Sustained or Quick Time Efforts or Strong or Light Weight Efforts. These models are then used for interactive sonifica-

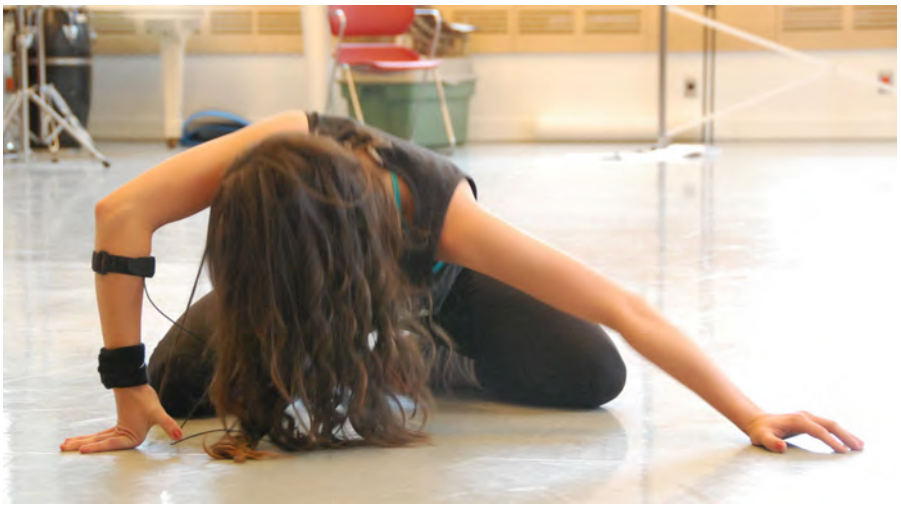


Figure 9.13: A participant equipped with the sensors: 3D accelerometer (wrist) and EMG (forearm).

tion: a dancer equipped with sensors (accelerometers and EMG) can control the re-synthesis of the CMAs' vocalizations (either Time or Weight Effort Factors).

9.4.4 Evaluation Workshop

We organized a one-day workshop with dance participants that learned Laban Efforts through interactive sonification. The workshop was facilitated by CMA *Donna Redlick*, and the dancers were observed by CMAs *Michelle Olson* and *Leslie Bishko*. The facilitator was asked to use the interactive sonification system to support the teaching of Effort Factors.

In the first session, the participants were given an overview of LMA and began to explore its structure through experiential sessions led by Donna Redlick. In the second session, participants were equipped with the set of sensors, and interacted with the sonic system. During both sessions they were guided by the facilitator, and their performance of the Effort Factors was observed and analyzed by two other CMAs. Observers and participants were encouraged to talk aloud about their experience or observations using their own terminology and including, when possible, LMA terminology. During the interactive session, each participant was guided by the facilitator to improvise with the sonic feedback in order to experience the targeted Effort factor and exhibit qualities at the extreme end of the Time and Weight Factors. Later, other participants could join and experience the targeted Effort Factor by attuning either to the movement of the equipped participant or to the produced sound.

PARTICIPANTS We recruited 5 participants (all female between 20 and 40 years old), with several years of dance practice, but no prior knowledge or practice of LMA. Participants were all comfortable with improvising and discussing their experience of movement exploration as well as being equipped with sensors and interacting with the sonic feedback.

DATA ANALYSIS The workshop was videotaped and the observations of the three CMAs were recorded. We used qualitative analysis for the transcriptions of the observations and discussions within the group in order to assess the participants' experience of the interactive sonic feedback and to capture the emerging guidelines for the design of sonic systems dedicated to movement training. We specifically focused on the comments relating to the effect of the interactive sound feedback on the participants' movement; the relationship between the movement, the sound, and the observed Effort Factors; and the experience of the interaction itself.

9.4.5 Results

We report here the main results that emerge from the qualitative analysis of open discussions among the group. We refer to participants as P1, P2, etc.; to observers as CMA1, CMA2; and to the facilitator as F. We use the terms 'mover' to designate the participant in interaction. All citations come from transcription.

MOVEMENT TO SOUND Open discussions brought out several issues about the effect of the interactive sonification on participants' movements, highlighting strengths and limitations in the design of the movement–sound relationship.

The sonification of the Weight Effort Factor was considered as responsive and consistent. Particular sounds were often revealed through specific movement patterns embodying the Weight Effort Factor: *"Now she [P2] is playing with that weight sense, and that contact [to the floor] gets the sound"*. Absence of Weight Effort Factor was also revealed through the sound, thus allowing to better access Weight Effort Factor: *"She went to vision drive and there was no sound. Vision Drive is weightless. It is Space, Time And Flow. And it was interesting because the sound stopped." (F).* [9]

Time Effort Factor sonification suffered from latency due to a filtering of the results intended to improve the recognition and smooth the sound feedback. Moreover, the relationship between Effort and sound was not perceived as transparent for Time Effort sonification. Several participants commented that the feedback contained much more information than Time only, often correlating it to Weight. These comments suggest the difficulty, highlighted by the CMAs during the recording sessions, to perform and vocalize Time Effort as an isolated Factor, and in that case to design movement–sound mappings.

9 LMA defined four Effort Drives combining three Effort factors and missing one Effort factor. Action Drives miss Flow; Spell Drives miss Time, Vision Drives miss Weight, and Passion Drives miss Space Effort.

Finally, inter-subject variability was brought up by observers who noticed different sound outcomes according to the mover's personal palette. Indeed some participants naturally accessed Time or Weight nuances more easily in their movement signature: *"I was hearing more sounds in this palette that I didn't hear in other people's movements."* (CMA2). This observation might correlate to the issue of sensor calibration, yet it also points to the nature of differing movement signatures elicited by each human body. In particular, while muscular activation highly varies from one participant to another, it also requires fine-tuning for each participant.

SOUND TO MOVEMENT Movers consistently reported an experience of attuning to the sound, often engaging in an exploration of movement directed towards the production of sound: *"I was trying to figure out how to make a sound, knowing that my body had the vocabulary to do it."* (P1). Besides, wearing the sensing devices themselves seemed to influence the participants' behavior, as reported by CMA2 who noticed a *"more isolated use of body parts"* of the equipped participant. In such cases the facilitator guided the movers towards an exploration of other body parts, often resulting in changes in the sound feedback: *"You were making some very new and interesting sound when initiating from the torso."* (P2). Finally, the sound feedback influenced the performance of Effort Factors. CMA2 reported on a portion of the interactive session during which all participants were improvising with Time Effort Factor: *"There was often very percussive sounds that I think were stimulating everyone to go into Quick Effort."*. Due to the ambiguities of Time Effort Factor sonification, the feedback could sometimes lead to changes in Effort that didn't relate to the sonified Effort Factor: *"what I didn't see in you moving before [before interacting with sound] is you increasing pressure. Adding weight to your vocabulary."* (F).

The CMAs and participants unanimously acknowledged the potential for new understanding, support for pedagogical opportunities afforded by technology, and the creation of a reflective space for learning.

9.4.6 Discussion and Conclusion

We have reported the results of a workshop intended to evaluate an approach to the sonification of Laban Effort Factors based on experts' vocalizations. The participants and experts had a very positive experience of the workshop and acknowledged its potential for supporting a better understanding of Effort Factors particularly within dance pedagogy. Several guidelines emerge from the discussions between participants and experts, providing precious insights for future development of such interactive sonic systems. First, the results stress the importance of tightening the relationship between movement and sound by limiting the latency and guaranteeing the transparency of the mapping between Effort Factors and sound. Technically, this requires a thorough selection of the training examples with a specific focus on quality, consistency, and alignment. In particular, the intrinsic difficulty of articulating the vocalization and performance

of isolated Effort Factors argues for the need of in-depth studies that correlate vocalizations' perceptive attributes with the identification of movement qualities. Finally, the very personal nature of Effort performance and experience questions the transferability of the Effort models among movers. This aspect motivates the development of higher–level movement features and richer computational models of Effort Factors that can adapt to a mover's personal signature by continuously learning during interaction.

Acknowledgments
We acknowledge support from the MovingStories project[10] and from the ANR project Legos 11 BS02 012. We thank CMAs Donna Redlick, Michelle Olson, Leslie Bishko, Philip Seth, and Karen Studd for their involvement in the workshop or the Motion Capture session.

9.5 Discussion

Humans use vocalizations in conjunction with gestures in a number of situations, from everyday communication to expert movement performance. For Mapping-by-Demonstration, vocalization is an efficient and expressive way to provide the system with sound examples that can be accurately performed along movements. Our system uses a temporal sequence model that allows to design the relationships between complex dynamic movements and sounds intuitively. The applications of the system range to sketching situations in sonic interaction design, to movement practice where it can be used to support pedagogy. Our exploratory experiments support the argument that continuous sonification can help learning movement dynamics, especially when visual feedback is inappropriate.

TOWARDS CONTINUOUS PARAMETRIC SYNTHESIS

Granular synthesis is a simple yet powerful technique for resynthesizing recorded sounds. Combined with the k-NN search inherited from CataRT, it provides a parametrized analysis-synthesis framework for sound textures. One of its critical limitation, however, lies in its *corpus-based* approach. If the corpus is only composed by the vocal demonstrations of the user, then the accessible palette of sound is restricted: it does not allow for interpolating nor extrapolating, even when the sound parameter estimated by the parameter mapping model vary from the demonstration.

The Master's thesis of Pablo Arias, that I supervised at Ircam, along with Norbert Schnell and Frédéric Bevilacqua, focused on improving interactive sound analysis/synthesis in a Mapping-by-Demonstration framework to move towards expressive, continuous parametric sound synthesis. We aimed to address the problem of *correspondence*, i.e. the perceptive match between the sounds used in demonstration and performance, through an improved temporal structure of the sound synthesis. We first proposed a

10 http://movingstories.ca/

transient conservation mechanism for granular synthesis. Then, we investigate a hybrid synthesis engine that combines additive synthesis with the granular engine with transient conservation. The main developments and results of Pablo Arias's Master's thesis are outlined in Appendix A.3.

FUTURE DIRECTIONS This work presents a first attempt to a design methodology based on vocal sounds. The most direct perspective concerns the relationship between the vocal imitations and the final sound synthesis. In the current approach, we chose to directly interact with the synthesis of the vocalizations. One of the most promising perspective of this work is to use vocal imitations only as a means to describe other types of sounds. The next step in the system design will therefore focus on making a link between the vocalizations and the final sound synthesis. For this purpose, we need to investigate more deeply what strategies people use to imitate sounds, and what strategy they use to associate movements to sounds and their imitation, as proposed in the Skat-VG project (Rocchesso et al., 2015).

9.6 Summary and Contributions

We proposed a generic system that learns the relationship between physical gestures and vocalizations. The system uses Hidden Markov Regression (HMR) to generate sound parameters representing the voice, from continuous movements. We presented an installation implementing Mapping-by-Demonstration for novice users, under the form of a game using gestural and vocal imitations. The quantitative analysis of novice users' performances highlight the high idiosyncrasy of gestures and vocalizations, and supports the idea that sound feedback helps learning motion dynamics. We reported another application of the system to movement pedagogy through an exploratory workshop with dancers. The system, trained with expert vocalizations and movements, was used to sonify Laban effort factors to support their apprehension by non expert dancers. Finally, we started investigating novel strategies for improving the temporal structure of the sound synthesis of vocalizations.

10

Conclusion

10.1 Summary and Contributions

The Mapping-by-Demonstration approach draws upon existing literature that emphasizes the importance of bodily experience in sound perception and cognition. Mapping-by-Demonstration is a framework 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.

Mapping-by-Demonstration combines the design principle of mapping through listening, that considers, at a higher level, listening as a starting point for mapping design, with interactive machine learning that allows for interaction-driven design.

We identified several key aspects of the parameter mapping layer, in particular multimodality and temporal modeling, that we addressed using probabilistic models. We proposed to fully exploit the generative nature of probabilistic models for mapping and sound parameter generation. While several regression methods have been proposed for learning the relationships between movement and sound parameters, few take a fully probabilistic perspective. Probabilistic models provide a fertile ground for continuous interaction as they allow for real-time, flexible, and parametric control of audio processing through parameter generation algorithms.

PROBABILISTIC MOVEMENT MODELS

We investigated probabilistic models for movement modeling, as a first iteration in the development of the Mapping-by-Demonstration framework. We proposed an implementation of three probabilistic movement models with varying levels of temporal modeling: Gaussian Mixture Models, Hidden Markov Models, and Hierarchical Hidden Markov Models. Our implementation emphasizes learning from few examples through user-defined regularization and complexity.

We formalized the mapping design patterns based on continuous gesture recognition with probabilistic models, discussing how likelihoods and

posterior probabilities can be exploited for sound control. We introduced a two-level Hierarchical Hidden Markov Model integrating a high level transition structure that improves continuous recognition. The model reinforces the temporal structure of the sound synthesis in Mapping-by-Demonstration (MbD) and allows users to author particular gesture representations.

PROBABILISTIC MULTIMODAL MODELS AND PARAMETER GENERATION

We implemented several models representing motion-sound sequences in a joint probabilistic framework, both instantaneous (Gaussian Mixture Regression (GMR)) and temporal (Hidden Markov Regression (HMR)). Our implementation is interaction-oriented and makes learning from few examples possible through user-authorable parameters such as regularization and complexity. We proposed a online inference algorithm for Hidden Markov Regression that continuously generates sound parameters given an input movement, and we derived a hierarchical extension of the model.

Jointly modeling motion and sound sequences provides a consistent representation of the variations occurring in both modalities, especially when coupled with a sequence model that takes advantage of contextual information. The probabilistic approach estimates the uncertainty over the synthesized sound parameters, consistently and in relation to the uncertainty of the input movement, which open novel possibilities for sound control.

APPLICATIONS AND EXPERIMENTS

We presented several concrete applications in movement performance, sonic interaction design, and dance. We proposed two approaches to movement analysis based on Hidden Markov Model and Hidden Markov Regression, respectively. We showed, through a use-case in Tai Chi performance, how the models help characterizing movement sequences across trials and performers.

We developed two generic systems exploiting probabilistic regression models. First, we created a system for crafting hand gesture control strategies for the exploration of sound textures, based on Gaussian Mixture Regression. Second, we exploited the temporal modeling of Hidden Markov Regression to develop a system associating vocalizations to continuous gestures. Both systems gave birth to interactive installations that we presented to a wide public, and we started investigating their interest to support gesture learning.

While we started working on vocalization as a use-case illustrating the possibilities of the models, many of the applications presented along this thesis take advantage of co-produced gestures and vocal sounds. Vocalization is advantageous in sound design for sound description and is widely used to support movement practice and performance. Therefore, vocalizations performed while moving appears as a promising perspective for further developments of the Mapping-by-Demonstration framework.

SOFTWARE By making the source code publicly available,[1] we also contribute to the dissemination of this scientific and technical knowledge to a wider community. Through this process we aim to 1) increase the reproducibility of the proposed research and its applications, 2) let other researchers use, improve and extend the source code to foster the development of novel applications, and 3) let musicians, composers, interpreters, hackers exploit and explore the possibilities of interactive machine learning systems for creative purposes. The proposed models are released as a portable, cross-platform c++ library that implements Gaussian Mixture Models and Hidden Markov Models for both recognition and regression. The *XMM* library was developed with interaction as a central constraint and allows for continuous, real-time use of the proposed methods for motion-sound Mapping-by-Demonstration.

10.2 Limitations and Open Questions

We formalized the concept of Mapping-by-Demonstration as "a framework for crafting sonic interactions from corporeal demonstrations of embodied associations between motion and sound; that uses an interactive machine learning approach to build the mapping from user demonstrations". We close this dissertation with a discussion on the limitations of the current approach, suggesting a number of both short-term and long-term perspectives.

HUMAN FACTORS In their review of robot Programming-by-Demonstration (PbD), Argall et al. (2009) identify two major causes for poor learning performances in a PbD framework, both related to the demonstration dataset: sparsity (the presence of undemonstrated states), and poor quality of the examples (task not performed optimally by the demonstrator). Obviously, both these flaws can be encountered in motion-sound mapping design, where users only demonstrate a limited set of examples, and might not always show a perfect consistency.

The issue of expertise is central to our applications: while the general idea of MbD underlines the intuitiveness of the approach, several aspects can be limiting for novice users. Obviously, some technical factors are at stake: the implementation might require programming, and the training procedure itself can be hard without a good understanding of the underlying mechanisms. Another aspect relates to the practice with the system, that is necessarily conditioned by the chosen mapping algorithm and by the type of sound synthesis. Learning to play with the constraints of the system is essential to become expert in providing 'efficient' and expressive demonstrations. For these reasons, it is crucial to develop a user-centered, interaction-driven implementation of the framework that enable users to quickly iterate in the design process.

1 **Note to reviewers:** The library has not yet been release but will be published before the defense.

TECHNICAL FACTORS A number of technical factors might also introduce errors or inconsistencies. First and foremost, the issue of correspondence implies non-trivial problems of feature extraction and sound synthesis, as discussed in Section 3.4. Second, learning the mapping between sequences of motion parameters and sound parameters is in itself a hard problem, especially when learning from limited data. Finding a compromise between overfitting and oversmoothing might be difficult in practice. Oversmoothing is a well-known problem in speech synthesis, and has been addressed by several improvements of the modeling techniques (Toda and Tokuda, 2007). In MbD, we might solve this issue by both human and technological factors. While it is possible to improve the models themselves, we believe that humans have a great adaptation abilities. For example, a human user might address undershooting by exaggerating the gestures during the performance phase in order to reach the desired sound result. As expertise increase, users might integrate this process in the design loop, and take into account the limitations of the recognition or generation algorithms to redesign their example gestures. Hence, we need to address the current limitation by integrating users in the loop with a fully interactive perspective on machine learning.

Finally, giving users the possibility to explore and generate new sounds and interactions is essential for expressivity. We need to furter investigate whether and how the MbD approach allows users both to reproduce, explore, and extend in performance the motion-sound relationships that were acted within the demonstrations.

MBD'S EMBODIMENT GAP Mapping-by-Demonstration aims to integrate action-perception more tightly into computational models for interaction design. We believe that the framework provides an interesting ground for studying embodied cognition phenomena. However, there exists an important pitfall in the current implementation of the framework, that we call MbD's Embodiment Gap. Mapping-by-Demonstration mirrors two views of the motion-sound relationships: the acted experience emerging from listening, and the interactive experience resulting from direct interaction with the learned system. There is an obvious gap between these experiences, that relates to the notions of agency and engagement (Leman, 2008). *Being in interaction* changes the way we perceive motion-sound interactions. While the demonstration approach might give an embodied way to design motion-sound relationships, it hardly fully accounts for what the experience of controlling the sound might be.

Nonetheless, an interesting adaptation phenomenon occurs in the current implementation. While at the first trial it might be difficult to apprehend or predict the experience of the actual interaction; by practicing, users learn the system's mechanisms, its limitations and possibilities. As expertise increases, users start to anticipate this gap between the simulated and actual interactions; and, along the trials, become able to integrate this gap into the design of their demonstrations.

Understanding this gap remains a challenge, and a long term goal would be to smooth the transition between demonstration and performance, taking their mutual influence into account until they consistently integrate.

10.3 Perspectives

FROM RECOGNITION TO CONTINUOUS VARIATIONS Along this thesis, we have emphasized the need to encode variations, both in terms of multiple classes of relationships, and as continuous variations within a single class. Encoding such variations is a difficult problem, as it requires to extrapolate from a very restricted set of examples while guaranteeing the consistency of the motion-sound relationship. Difficulties emerge both from the very definition of such 'consistency', that is ambiguous and context-dependent, and from the technical issue of generalizing from few examples. While we believe that joint multimodal models of motion and sound better encode these variations, further research is needed to understand how we perceive and exploit such variations in motion and sound. Speech processing and robotics aggregate significant expertise in probabilistic modeling, that might be of interest for future developments in our field. For example, several approaches have been developed for adapting models to new users (Speaker Adaptation (Leggetter and Woodland, 1995)) or to contextual factors (e.g. , emotion (Ding, 2014)); or, in robotics, for combining the constraints defined by several demonstrations (Calinon, 2007).

EXPERTISE Mapping-by-Demonstration implements a design process driven by the action-perception loop that aims at supporting intuitiveness in the creation of sonic interactions. However, as discussed above, expertise is essential at technological and usability levels. Today, significant programming is still required for developing new systems based on the proposed machine learning algorithms. The contributions of this thesis target users at two different levels: expert users able to program and manipulate the learning algorithms, and novice users interacting through installations. Addressing a wider range of users — in particular expert musicians, artist or hackers that do not necessarily have an extensive knowledge of machine learning, — demands further developments in the workflow and implementation. We believe that getting users to understand the internal behavior of the computational model is essential to design expressive interactions, and might be supported by interactive visualizations of the models' internal structure.

Sketching Sonic Interactions with Vocalizations and Gestures

Along this thesis, vocalization in movement practice has become a primary interest, that opens novel perspective for movement and sound design. We proposed an application of the MbD framework where the demonstrations are produced vocally by the user while moving. In performance, our system (re-)generates vocalizations interactively from the user's movements.

We now consider a broader application where the vocalizations of the demonstration phase mimic or imitate the sounds to be realized in performance. This introduces a highly complex correspondence between the sound space defined by the demonstration and the sounds to synthesize in performance. It is worthwhile noting that the problem of correspondence, in this case, it both technological (*how to remap one corpus over another?*), and human (*how do we vocally imitate sounds?*). Therefore, it is essential to identify both the strategies used in vocal imitations of sounds, and the strategies used to associate body movements to such imitations, as proposed, for example, in the Skat-VG project (Rocchesso et al., 2015).

A

Appendix

A.1 Publications

This section presents the published contributions and outlines their relationship with the work presented in this dissertation.
The full text of all publications can be downloaded from:
http://julesfrancoise.com/publications/

As First Author

J. Françoise, N. Schnell, R. Borghesi, and F. Bevilacqua, **Probabilistic Models for Designing Motion and Sound Relationships**," in *Proceedings of the 2014 International Conference on New Interfaces for Musical Expression*, ser. NIME'14, London, UK, 2014, pp. 287–292.

> This article presents the XMM library and its Max implementation. It reports a very short outline of the four models studied in this dissertation, each of which is illustrated with a use case in sound control. Parts of this article are reported in Chapter 3 and Appendix A.2.

J. Françoise, N. Schnell, and F. Bevilacqua, **MaD: Mapping by Demonstration for Continuous Sonification**," in *ACM SIGGRAPH 2014 Emerging Technologies*, ser. SIGGRAPH '14. Vancouver, Canada: ACM, 2014, pp. 16:1—-16:1.

> This paper supports a demonstration proposal for the ACM SIGGRAPH Conference. We proposed two installations: the system allowing to interact with environmental sounds from hand movements presented in Section 8.3 and the imitation game reported in Section 9.3.

J. Françoise, S. Fdili Alaoui, T. Schiphorst, and F. Bevilacqua, **Vocalizing Dance Movement for Interactive Sonification of Laban Effort Factors**," in *Proceedings of the 2014 Conference on Designing Interactive Systems*, ser. DIS '14. Vancouver, Canada: ACM, 2014, pp. 1079–1082.

> We investigate the use of interactive sound feedback for dance pedagogy based on the practice of vocalizing while moving. Specifically, the paper proposes an approach for learning mapping strategies from expert performances and vocalizations. We investigate the sonification of Laban Effort Factors in an exploratory workshop with dancer. The totality of this article is reported in Section 9.4.

J. Françoise, N. Schnell, and F. Bevilacqua, **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, 2013, pp. 705–708.

> We propose the use of the multimodal HMM for learning the relationships between motion and sound. We also propose an application to the control of physical modeling sound synthesis. We present in more detail the formalism and applications of HMMs for regression in Chapter 6.

J. Françoise, B. Caramiaux, and F. Bevilacqua, **A Hierarchical Approach for the Design of Gesture-to-Sound Mappings**," in *Proceedings of the 9th Sound and Music Computing Conference*, Copenhagen, Denmark, 2012, pp. 233–240.

> In this paper we outline the recognition gesture process based on the hierarchical HMM. We propose a mapping strategy that draws upon a segmental representation of gestures in four phases: Preparation-Attack-Sustain-Release. The contributions of this article are presented in Section 4.6.

J. Françoise, **Gesture–Sound Mapping by Demonstration in Interactive Music Systems**," in *Proceedings of the 21st ACM international conference on Multimedia (MM'13)*, Barcelona, Spain, 2013, pp. 1051—-1054.

> This doctoral symposium paper synthesizes the modeling approaches based on the hierarchical HMM and the multimodal HMM for gesture-sound mapping. The paper received the ACM Multimedia 2014 Best Doctoral Symposium Award.

J. Françoise, N. Schnell, and F. Bevilacqua, **Gesture-based control of physical modeling sound synthesis**," in *Proceedings of the 21st ACM international conference on Multimedia (MM'13)*. Barcelona, Spain: ACM Press, 2013, pp. 447–448.

> This short paper is a demonstration proposal supporting the previous publication. It uses the multimodal HMM to learn the mapping between movements and physical modeling sound synthesis. The totality of this use-case is reported in Section 6.3.

J. Françoise, I. Lallemand, T. Artières, F. Bevilacqua, N. Schnell, and D. Schwarz, **Perspectives pour l'apprentissage interactif du couplage geste-son**," in *Actes des Journées d'Informatique Musicale (JIM 2013)*, Paris, France, 2013.

> This paper presented prospective work on combining active learning from human reward with the hierarchical HMM.

J. Françoise, **Realtime Segmentation and Recognition of Gestures using Hierarchical Markov Models**," Master's Thesis, Université Pierre et Marie Curie, Ircam, 2011.

This dissertation details the study and linear time implementation of the hierarchical HMM for gesture segmentation and recognition. The technical details of the model are partly reported in Section 4.5.

As Secondary Author

F. Bevilacqua, N. Schnell, N. Rasamimanana, J. Bloit, E. Fléty, B. Caramiaux, J. Françoise, and E. Boyer, **De-MO : Designing Action-Sound Relationships with the MO Interfaces** in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, Paris, France, 2013.

This extended abstract proposes a demonstration at CHI'13 *Interactivity* track of the Modular Musical Objects (MO), an ensemble of tangible interfaces and software modules for creating novel musical instruments or for augmenting objects with sound. We demonstrated a use case with Hidden Markov Regression that implemented continuous gesture recognition and mapping to physical modeling sound synthesis.

B. Caramiaux, J. Françoise, N. Schnell, and F. Bevilacqua, **Mapping Through Listening**, *Computer Music Journal*, vol. 38, no. 3, pp. 34–48, 2014.

This paper describes a general methodology integrating perception-action loop as a fundamental design principle for gesture–sound mapping in digital music instruments. Our approach considers the processes of listening as the foundation — and the first step — in the design of action-sound relationships. In this design process, the relationship between action and sound is derived from actions that can be perceived in the sound. Building on previous works on listening modes and gestural descriptions we proposed to distinguish between three mapping strategies: instantaneous, temporal, and metaphoric. Our approach makes use of machine learning techniques for building prototypes, from digital music instruments to interactive installations. Four different examples of scenarios and prototypes are described and discussed. This paper is outline in the related work (Chapter 2). I contributed to the writing of the article, which reports a use-case using Hierarchical HMMs.

A.2 The XMM Library

We released a portable, cross-platform C++ library that implements Gaussian Mixture Models and Hidden Markov Models for both recognition and regression. The *XMM* library was developed with interaction as a central constraint and allows for continuous, real-time use of the proposed methods. The library is open source, available under the GNU General Public License (GPLv3):

https://github.com/Ircam-RnD/xmm

The models are also integrated with the *MuBu* environment within *Cycling 74 Max* that 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 provides a fluid workflow for recording, training and evaluating of the models, that we started complementing with a set of visualizations of the models parameters. MuBu is freely available on Ircam's *Forumnet*.[1]

By making the source code publicly available, we aim to contribute to the dissemination of this scientific and technical knowledge to a wider community. Through this process we aim to

- Increase the reproducibility of the proposed research and its applications.
- Let other researchers use, improve and extend the source code to foster the development of novel applications.
- Let musicians, composers, interpreters, hackers exploit and explore the possibilities of interactive machine learning systems for creative purposes.

Note:
Parts of this section are extracted from our article "Probabilistic Models for Designing Motion and Sound Relationships" presented at the NIME'14 International Conference on New Interfaces for Musical Expression (Françoise et al., 2014a).

A.2.1 Why another HMM Library?

Several general machine learning toolkits have become popular over the years, such as Weka[2] in Java, Sckits-Learn[3] in Python, or more recently MLPack[4] in C++. However, none of the above libraries were adapted for the purpose of this thesis. As a matter of fact, most HMM implementations are oriented towards classification and they often only implement offline inference using the Viterbi algorithm.

In speech processing, the Hidden Markov Model Toolkit (HTK)[5] has now become a standard in Automatic Speech Recognition, and gave birth to a branch oriented towards synthesis, called HTS.[6] Both libraries present many features specific to speech synthesis that do not yet match our use-cases in movement and sound processing, and have a really complex structure that does not facilitate embedding.

Above all, we did not find any library explicitly implementing the Hierarchical Hidden Markov Model (HHMM), nor the regression methods based on GMMs and HMMs. For these reasons, we decided to start of novel implementation of these methods with the following constraints:

1 MuBu on Ircam Forumnet: http://forumnet.ircam.fr/product/mubu/
2 http://www.cs.waikato.ac.nz/ml/weka/
3 http://scikit-learn.org/
4 http://www.mlpack.org/
5 http://htk.eng.cam.ac.uk/
6 http://hts.sp.nitech.ac.jp/

REAL-TIME Inference must be performed in continuously, meaning that the models must update their internal state and prediction at each new observation to allow continuous recognition and generation.

INTERACTIVE The library must be compatible with an interactive learning workflow, that allows users to easily define and edit training sets, train models, and evaluate the results through direct interaction. All models must be able to learn from few examples (possibly a single demonstration).

PORTABLE In order to be integrated within various software, platforms, the library must be portable, cross-platform, and lightweight.

We chose C++ that is both efficient and easy to integrate within other software and languages such as Max and Python. We now detail the four models that are implemented to date, the architecture of the library as well as the proposed Max/MuBu implementation with several examples.

A.2.2 Four Models

The implemented models are summarized in Table A.1. Each of the four model addresses a different combination of the multimodal and temporal aspects. We implemented two instantaneous models based on Gaussian Mixture Models and two temporal models with a hierarchical structure, based on an extension of the basic Hidden Markov Model (HMM) formalism.

	Movement	Multimodal
Instantaneous	Gaussian Mixture Model (GMM)	Gaussian Mixture Regression (GMR)
Temporal	Hierarchical Hidden Markov Model (HHMM)	Multimodal Hierarchical Hidden Markov Model (MHMM)

Figure A.1: Summary of the probabilistic models.

GAUSSIAN MIXTURE MODELS (GMMS) are instantaneous movement models. The input data associated to a class defined by the training sets is abstracted by a mixture (i.e. a weighted sum) of Gaussian distributions. This representation allows recognition in the *performance* phase: for each input frame the model calculates the likelihood of each class (Figure A.2**(a)**).

GAUSSIAN MIXTURE REGRESSION (GMR) Sung (2004) are a straightforward extension of Gaussian Mixture Models used for regression. Trained with multimodal data, GMR allows for predicting the features of one modality (e.g. sound) from the features of another (e.g. movement) through non-linear regression between both feature sets (Figure A.2**(b)**).

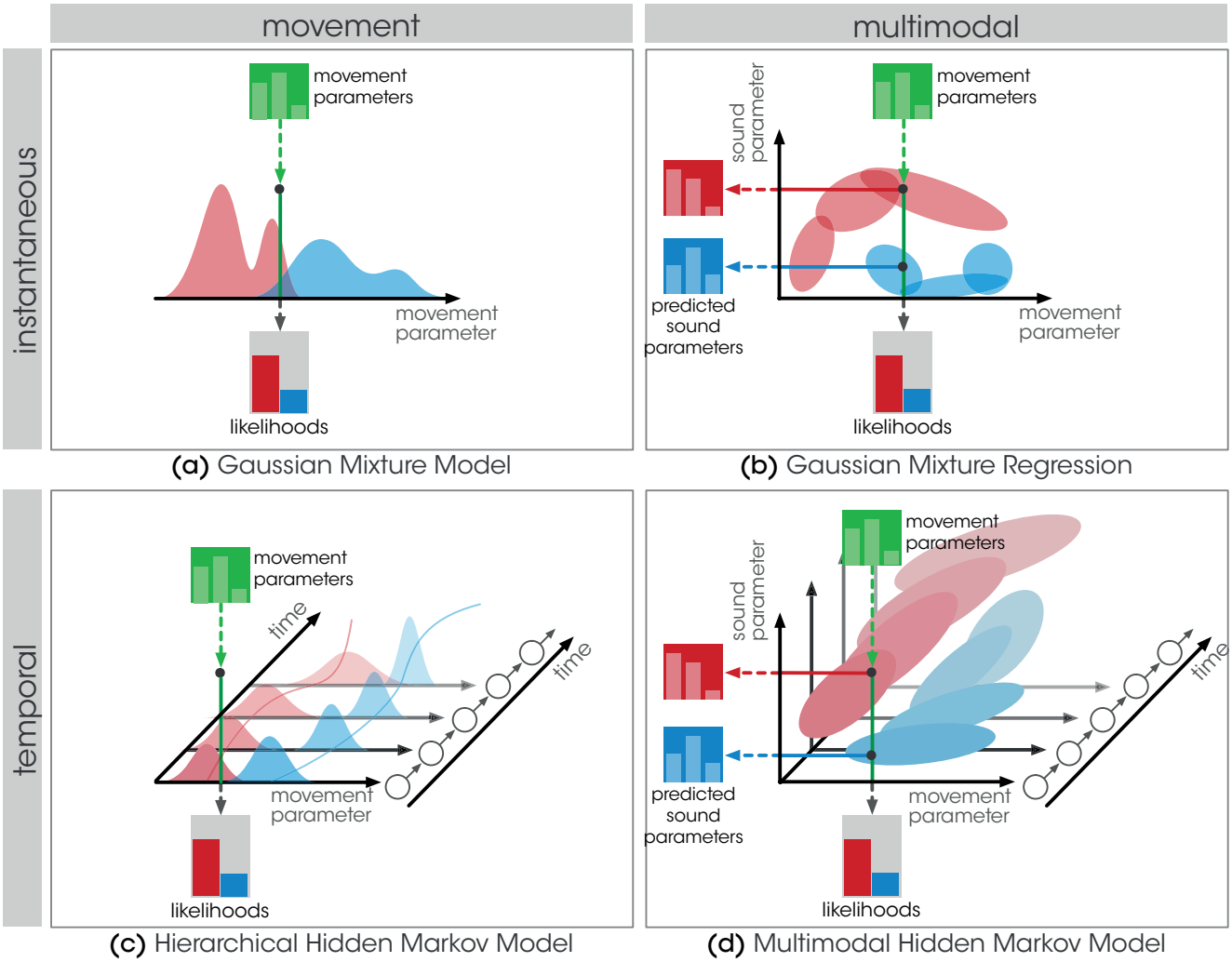


Figure A.2: Schematic representation of the characteristics of the 4 models.

HIERARCHICAL HMM (HHMM) Françoise et al. (2012) integrates a high-level structure that governs the transitions between classical HMM structures representing the temporal evolution of — low-level — movement segments. In the *performance* phase of the system, the hierarchical model estimates the likeliest gesture according to the transitions defined by the user. The system continuously estimates the likelihood for each model, as well as the time progression within the original training phrases (Figure A.2**(c)**).

MULTIMODAL HIERARCHICAL HMM (MHMM) Françoise et al. (2013b) allows for predicting a stream of sound parameters from a stream of movement features. It simultaneously takes into account the temporal evolution of movement and sound as well as their dynamic relationship according to the given example phrases. In this way, it guarantees the temporal consistency of the generated sound, while realizing the trained temporal movement-sound mappings (Figure A.2**(d)**).

A.2.3 Architecture

Our implementation follows the workflow presented in Chapter 3 with a particular attention to the interactive training procedure, and to the respect of the real-time constraints of the *performance* mode. The library is built upon four components representing phrases, training sets, models and model groups, as represented on Figure A.3. A phrase is a multimodal data container used to store training examples. A training set is used to aggregate phrases associated with labels. It provides a set of function for interactive recording, editing and annotation of the phrases. Each instance of a model is connected to a training set that provides access to the training phrases. Performance functions are designed for real-time usage, updating the internal state of the model and the results for each new observation of a new movement. The library is portable and cross-platform. It defines a specific format for exchanging trained models, and provides Python bindings for scripting purpose or offline processing.

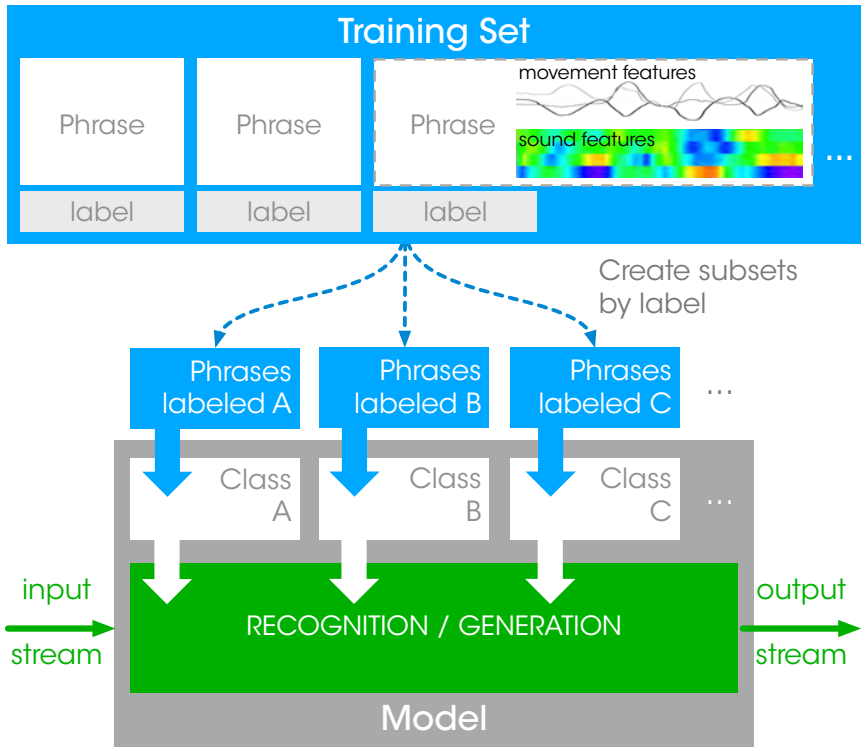


Figure A.3: Architecture of the XMM library.

A.2.4 Max/MuBu Integration

Max is a visual programming environment dedicated to music and interactive media. We provide an implementation of our library as a set of Max externals and abstractions articulated around the *MuBu*[7] collection of objects developed at Ircam (Schnell et al., 2009).

7 http://forumnet.ircam.fr/product/mubu/

Training sets are built using *MuBu*, a generic container designed to store and process multimodal data such as audio, motion tracking data, sound descriptors, markers, etc. Each training phrase is stored in a buffer of the container, and movement and sound parameters are recorded into separate tracks of each buffer. Markers can be used to specify regions of interest within the recorded examples. Phrases are labeled using the markers or as an attribute of the buffer. This structure allows users to quickly record, modify, and annotate the training examples. Training sets are thus autonomous and can be used to train several models.

Each model can be instantiated as a max object referring to a MuBu container that defines its training set. For training, the model connects to the container and transfers the training examples to its internal representation of phrases. The parameters of the model can be set manually as attributes of the object, such as the number of Gaussian components in the case of a GMM, or the number of states in the case of a HMM. The training is performed in background.

For performance, each object processes an input stream of movement features and updates the results with the same rate. For movement models, the object output the list of likelihoods, complemented with the parameters estimated for each class, such as the time progression in the case of a temporal model, or the weight of each Gaussian component in the case of a GMM. For multimodal models, the object also outputs the most probable sound parameters estimated by the model, that can be directly used to drive the sound synthesis.

A.2.5 Example patches

The applications described in this section are distributed as Max patches with the current release of the Max library on Ircam Forumnet.

RESONANT SCRATCHING This application aims at sonifying touching movements using a set of resonant models.[8] The application is depicted in Figure A.4, and a screenshot of the Max patch is reported in Figure A.5.

Motion capture is performed using a contact microphone placed on the control surface. Our goal is to classify different touching modes from the audio signal in order to select the separate resonant model. This classification only requires the instantaneous description of the timbre of the scratching sound. Therefore, we do not consider the temporal dynamics in this case, which justifies the use of an instantaneous movement model. We use a GMM to classify touch using Mel-Frequency Cepstral Coefficients (MFCCs), that we consider here as movement features since they directly relate to touch qualities.

During *Training*, we demonstrate several examples of 3 classes of touch: for instance *rub*, *scratch* and *tap*, by recording and analyzing the sound of each touching mode. Each class is represented by a GMM with 3 Gaus-

8 This application draws from previous research from the Interlude project (see: http://interlude.ircam.fr/) Rasamimanana et al. (2011)

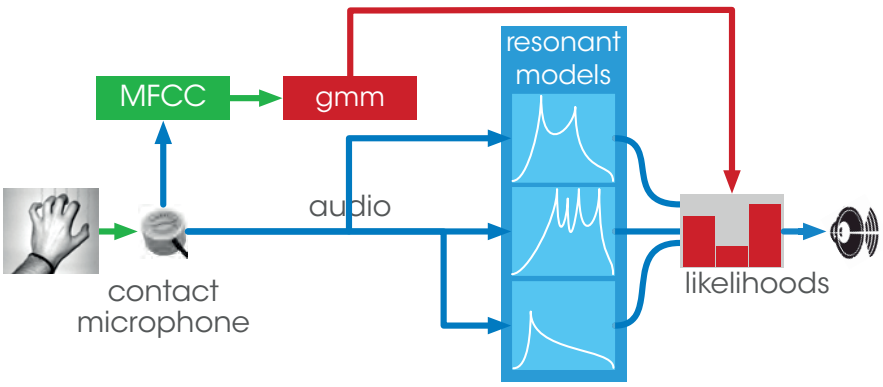


Figure A.4: Workflow of the *performance* phase of the *Scratching* application.

sian components, and is associated with a resonant model. During *Performance*, the sound from the contact microphone is then directly filtered using the resonant model. The amount of each filter is determined by the likelihood of each class.

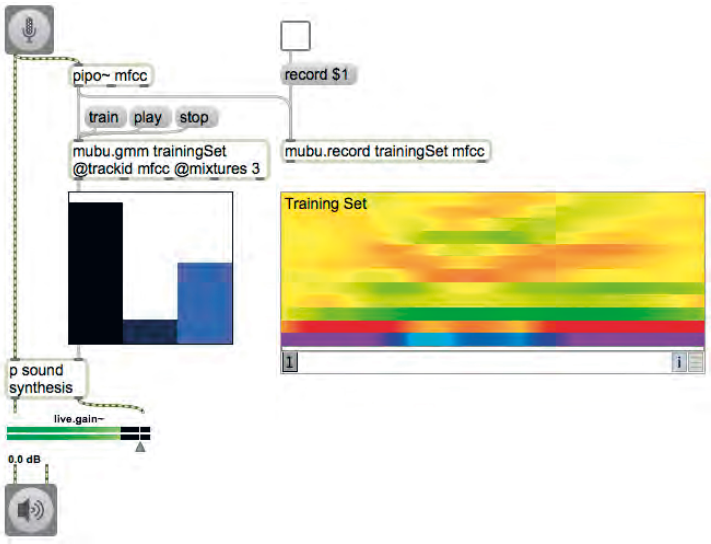


Figure A.5: Screenshot of the Max patch of the *Scratching* application.

PHYSICAL SOUND DESIGN In this application, we map in-air movement to physical modeling sound synthesis, as shown in Figure A.6**(b)**). Using a Leapmotion™hand tracking system, hand speed and orientation are directly available as movement features. The goal here is to learn the mapping between these movement features and the control parameters of physical models. Therefore, this application requires an instantaneous multimodal model, namely GMR.

For *Training*, we start by designing sounds using a graphical editor that allows us to draw time profiles of the physical models' input parameters. After recording several examples of movements with each preset, one model is trained for each physical model using movement and sound parameters sequences. During Performance, the GMR generates the control

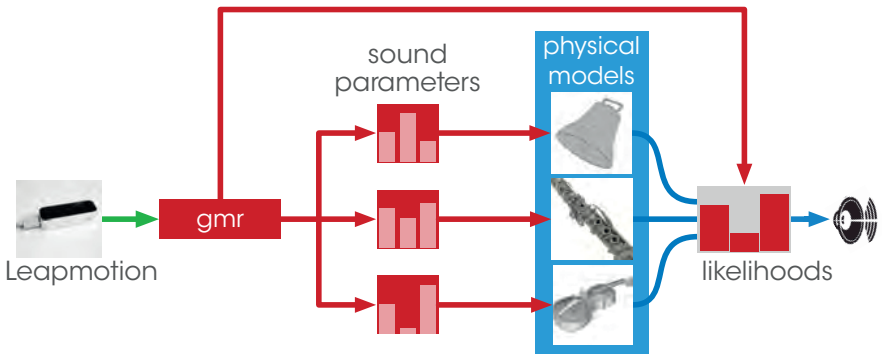


Figure A.6: Workflow of the *performance* phase of the *Scratching* application.

parameters of each physical models, and estimates the likelihoods, that are used to mix the sound output of each synthesizer.

GESTURE-BASED SOUND MIXING

This use case illustrates the use of the continuous estimation of the likelihoods in gesture recognition (Figure A.7**(c)**). The goal is to continuously control the mixing of a set of recorded sounds, from a set of dynamic gestures captured using the Leapmotion. As dynamic gesture recognition is required here, we use a temporal movement model, namely a HHMM.

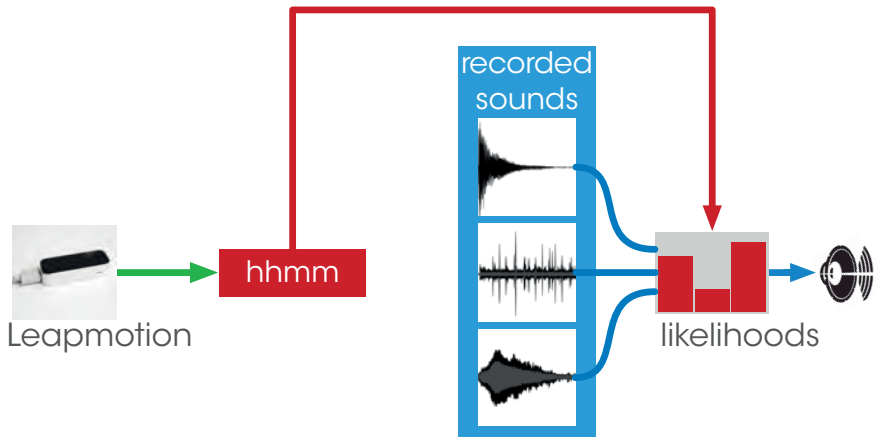


Figure A.7: Workflow of the *performance* phase of the *Scratching* application.

After defining the gesture vocabulary, we record several examples of each gesture to recognize, taking care of varying particular aspects such as the speed and breadth of each movement to ensure generalization and robustness of the recognition method. The movement models are learned using a HHMM in which each sub-model represents a particular class of gesture. As shown in Figure A.7**(c)**, during performance the HHMM is used to evaluate the likelihood of each gesture, that is used to drive the playback level of the associated recorded sound.

INTERACTIVE VOCALIZATION This prototype focuses on sonic interaction design based on movements and non-verbal vocal sketches (Figure A.8**(d)**). The application allows for performing interactive vocalizations where the relationships between motion and sounds are learned from direct demonstration of movements and vocalizations performed synchronously during the *training* phase. Movements are captured using MO interfaces (Rasamimanana et al., 2011), that integrate 3D accelerometers and gyroscopes. In order to guarantee a consistent reconstruction of the vocal sketches, this application requires the use of a temporal model. Therefore, we use the MHMM model to learn this multimodal and temporal mapping.

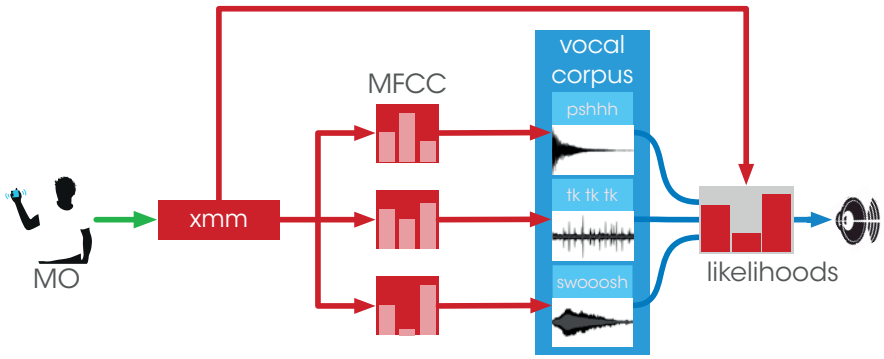
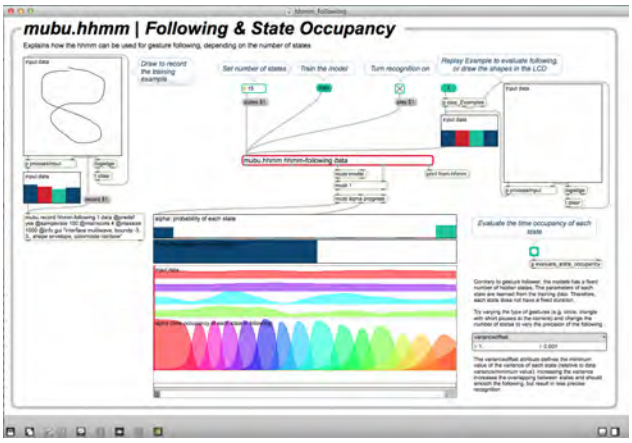


Figure A.8: Workflow of the *performance* phase of the *Scratching* application.

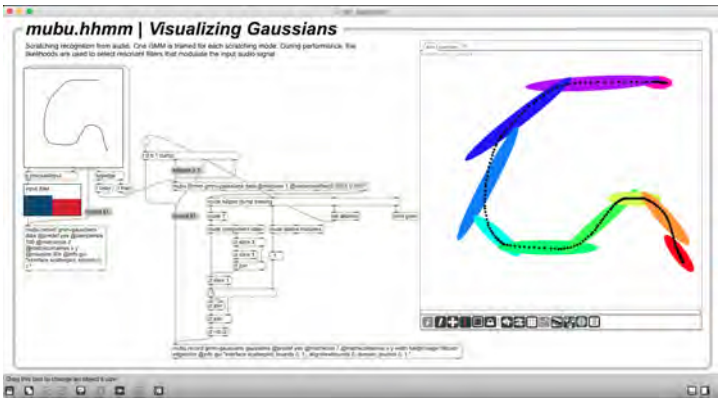
Each training phrase associates a sequence of motion features with a sequence of MFCCs computed from the audio. From this multimodal data, a hierarchical model (MHMM) is learned, in which each sub-model represents a multimodal primitive linking movement and voice. During *performance*, the model recognizes the movement and estimates the MFCCs accordingly. We use a corpus-based granular synthesis engine. The estimated stream of MFCCs is used to re-synthesize the vocalizations by concatenating the grains that match the sound description using a KNN search (Schwarz, 2007). As before, the likelihoods are used to control the level of each class of vocalization.

A.2.6 Future Developments

VISUALIZATION We have started integrating several tools for visualizing the models' parameters and recognition process. For example, we provide an example patch allowing to visualize the distribution of time spent on each state in gesture following, which is presented in Figure A.9a. We plan to integrate further visualizations of the models' internal parameters, notably through the representation of the Gaussian parameters (states or mixtures) as confidence interval ellipses. Such representations might help users understand the behavior of models and allow them to optimize the training.



(a) Visualization of state time occupancy



(b) Visualization of Gaussian parameters

Figure A.9: Examples of prototype visualizations of the models' parameters

A.2.7 Other Developments

LEAP MOTION SKELETAL TRACKING IN MAX

Several other developments were achieved for experimentation and research purposes. Notably, we developed a Max object interfacing with the Leap Motion controller that implements most of the features of the Leap Motion SDK V2 — in particular hand and finger identification, full skeletal tracking, — that were not available to date in the Max environment. The objects comes with record/play and visualization utilities using the MuBu environment (see the screenshot of Figure A.10). The object is open-source and publicly available on Ircam-Forge.[9]

Figure A.10: Screenshot of the Leap Motion Skeletal Tracking Max external

9 Leap Motion for Max: http://forge.ircam.fr/p/leapmotion/

BAYESIAN FILTERING FOR EMG ENVELOPE EXTRACTION

We also developed a C++ library and Max external for (Electromyogram) EMG envelope extraction based on Bayesian Filtering (Sanger, 2007). The method addresses the limitations of low-pass filtering techniques that suffer from a smoothing of rapid changes in the EMG signal. Here, "the filtered signal is modeled as a combined diffusion and jump process, and the measured EMG is modeled as a random process with a density in the exponential family. [...] This estimate yields results with very low short-time variation but also with the capability of rapid response to change." (Sanger, 2007). The method is integrated within the PiPo[10] framework associated to the MuBu collection (see the screenshot of Figure A.11).

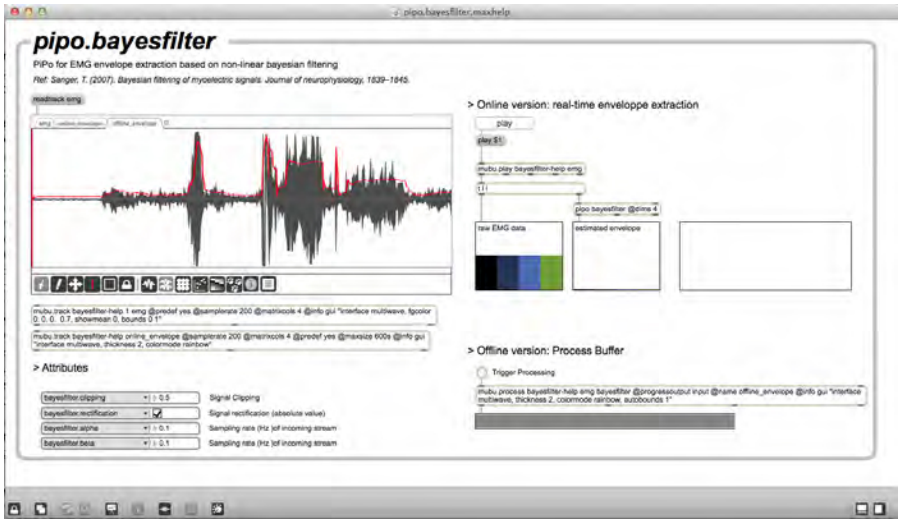


Figure A.11: Screenshot of the help patch of the EMG envelope extraction external `pipo.bayesfilter`

A.3 Towards Continuous Parametric Synthesis

This section outlines the main developments and results of Pablo Arias's Master's thesis. I supervised Pablo Arias' internship at Ircam, along with Norbert Schnell and Frédéric Bevilacqua. His internship focused on improving interactive sound analysis-synthesis for Mapping-by-Demonstration, with vocalization as a primary use-case. For more details, see the full Master's Thesis: P. Arias, "Description et synthèse sonore dans le cadre de l'apprentissage mouvement-son par démonstration," Master's Thesis, Ircam—Université Pierre et Marie Curie, 2014. (Arias, 2014).

10 http://ismm.ircam.fr/pipo/

A.3.1 Granular Synthesis with Transient Preservation

MOTIVATION Our system for gesture interaction with vocalization allows users to interact with vocal sounds in a non-linear way. In order to maximize the correspondence between the vocal sounds created in demonstration and the synthesis of the vocalizations in performance, we propose to improve the temporal structure of granular synthesis.

Granular synthesis can create artifacts on sounds presenting sharp transients. When several grains containing an attack are overlapped, the transient is replayed several times and can generate noisy sounds. We propose to integrate transient preservation in granular synthesis.

Our goal is to improve the *sound correspondence*, by maximizing the perceptive match between the vocal sounds created in demonstration and the synthesis of the vocalizations in performance. While phase vocoding methods already integrate such transient conservation, we focus on granular synthesis to keep an expressive control on the textural qualities of the sound, to ensure a low computational load, and to keep a consistent parametrization with the descriptor-driven approach.

PROPOSED METHOD The method is based on a preliminary annotation of the sound using attack detection. At runtime, the parameters of the granular synthesis are dynamically modified to playback the attacks without overlap, as illustrated in Figure A.12. The duration of

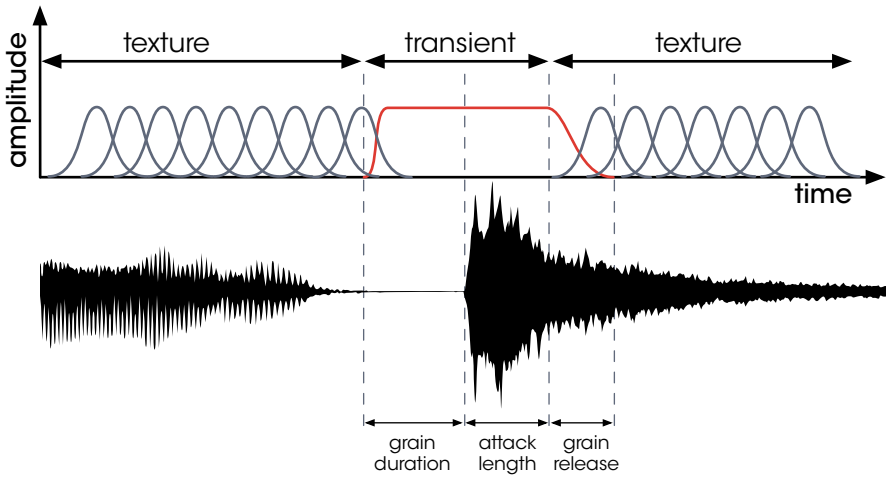


Figure A.12: Granular synthesis with transient preservation. Granular Parameters are dynamically modified to play attacks in a single grain without overlap.

the grain is adapted to the length of the attack, and begins one standard grain duration before the attack — to avoid that previous grains contain a part of the transient. We add a long release to the attack grain in order to smoothly fade into the 'texture' setting of the granular synthesis.

A video demonstration of the system, used in a Mapping-by-Demonstration (MbD) system with a graphic tablet as in put, is available online[11].

A.3.2 A Hybrid Additive/Granular Synthesizer

As a first step towards a parametric synthesis engine, we consider additive synthesis for continuous control of the voiced part of vocalizations. We propose an implementation of the harmonic plus noise model that combines additive synthesis with the granular engine with transient conservation, that gives a higher level of control on the textural qualities of the residuals.

OVERVIEW Additive synthesis is based on the representation of periodic signals as a sum of sinusoidal components multiple of the fundamental frequency. Many musical and vocal signals can be considered as pseudo-periodic and can therefore be represented as a time-varying sequence of partial amplitude, frequency and phase. Notably, Serra (1989) proposed a model composed of deterministic and stochastic components that can be modeled by a harmonic part and residual noise:

$$y(t) = \sum_{k=1}^{K} A_k \cos\left[\theta_k(t)\right] + e(t) \tag{A.1}$$

where K is the number of partials, A_k and θ_k are the amplitude and instantaneous phase of the kth partial, and $e(t)$ is the residual noise. MuBu embeds an additive synthesis engine based on the FFT^{-1} synthesis method proposed by Rodet and Depalle (1992). The additive part can be controlled using the amplitude, frequency and phase of each partial, while the residuals are controlled using a time index. The quality of the synthesis is limited by the poor rendering of the residuals using the FFT^{-1} method.

We propose to integrate the additive synthesis with a more flexible engine for the residuals, based on a modified version of granular synthesis. The overall analysis-synthesis process is outlined in Figure A.13. The sound is analyzed by Ircam's *PM2*[12] (Partial Manager 2) that performs partial tracking Depalle et al. (1993). The resulting sequence of partials and residual noise is stored in a MuBu container. In parallel, we perform attack detection on the original sound to identify and annotate the transients. Sounds can be directly synthesized from a time index, that simultaneously scrubs into the sequence of partials and the residual audio track. The residuals are synthesized using a granular synthesizer with transient preservation.

A.3.3 Integration in a Mapping-by-Demonstration System

We experimented the integration of the sound synthesis in a MbD system mapping surface gestures to vocalizations. The gesture is described using

11 http://julesfrancoise.com/phdthesis/#synthesis
12 http://anasynth.ircam.fr/home/software/pm2

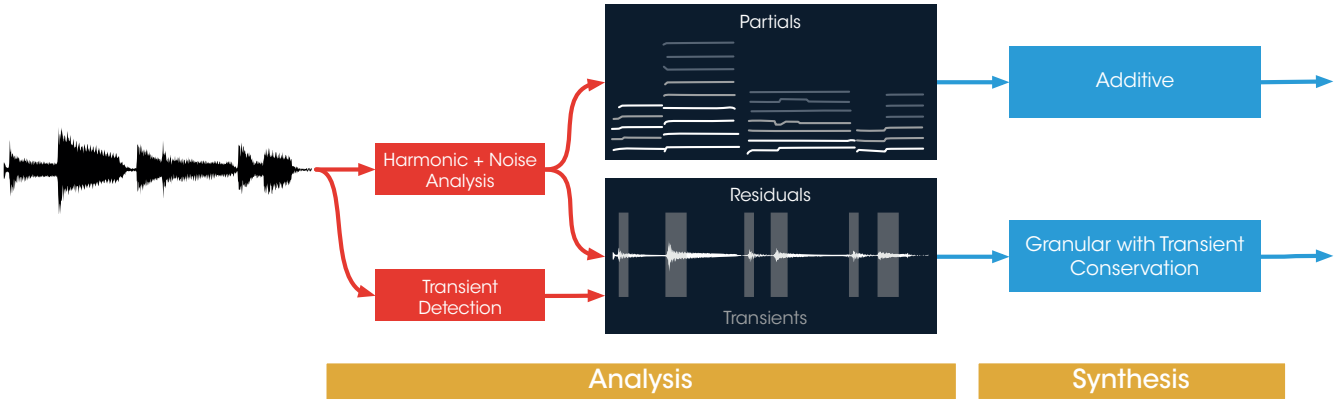


Figure A.13: Hybrid Additive/Granular analysis and synthesis.

the two-dimensional position from a Wacom tablet. Once the gesture and its associated vocalization have been recorded, the sound is analyzed to extract the sequence of partials and the resulting residuals.

We experimented with four mapping strategies based on Hidden Markov Regression (HMR), with the various synthesis engines:

GRANULAR is the initial approach, as described in Section 9.2.2. We learn a Hidden Markov Regression (HMR) between the sequences pf movement features and Mel-Frequency Cepstral Coefficients (MFCCs). The synthesis is performed with descriptor-driven granular synthesis.

GRANULAR+ATTACKS extends the previous approach to the synthesis engine with transient conservation. Similarly, we learn a HMR between the sequences of movement features and MFCCs. The synthesis is performed with descriptor-driven granular synthesis with transient conservation.

HYBRID FREQ&AMP uses the hybrid synthesis engine. We learn a HMR between motion features and the frequency and amplitudes of K partials. In performance, the HMR directly generates the partials amplitude and frequency to control the additive synthesize. The synthesis of the residual is controled by a time index estimated using the time progression of the Hidden Markov Model (HMM).

HYBRID F0&AMP uses the hybrid synthesis engine with a harmonic assumption. We learn a HMR between motion features and the frequency and amplitudes of K partials. In performance, the HMR directly generates the partials amplitude and frequency to control the additive synthesize. The synthesis of the residual is controled by a time index estimated using the time progression of the HMM.

A video that demonstrates the four synthesis engines is available online[13].

13 http://julesfrancoise.com/phdthesis/#synthesis

A.3.4 Qualitative Evaluation

METHOD We conducted a subjective evaluation with 10 participants from Ircam to assess the improvements of the sound synthesis in comparison with the original system using granular synthesis. The experiment was divided in two phases: the reproduction of gestures associated with vocal sounds according to predefined mappings, and the creation of vocalizations associated with new gestures. The participants were informed that the synthesis methods were under investigation, but were not aware of the difference between the synthesis engines. For each reference gesture-sound mapping, the participants were asked to assess the quality of the sound synthesis on a five-point Likert scale, given their experience with the interactive system.

RESULTS The average score and standard deviation are plotted in Figure A.14 for the 4 synthesis engines — 30 scores are given for each synthesis method. The scores increases with the successive improve-

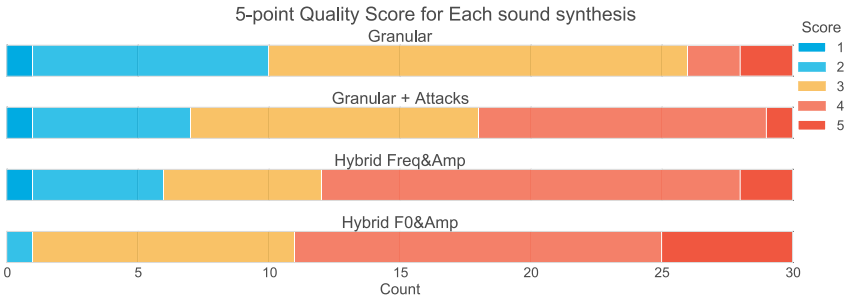


Figure A.14: 5-point Likert Scale scores of the subjective assessment of the quality of sound syntheses.

ments to the original sound synthesis. Participants appreciated transient conservation when using vocalization with sharp attacks. Globally, the hybrid synthesizer combining additive and granular synthesis with transient conservation was higher rated than the original method.

A.3.5 Limitations and Future Developments

The developments of the sound synthesis offer new possibilities for sound control. In particular, it allows us to investigate how probabilistic models of the mapping can interpolate or extrapolate from the given demonstrations, which tackles the wider issue of generating novelty while guaranteeing consistency. We have started investigating how such continuous parametric synthesis allowed to interpolate between vocal qualities. The idea is to learn a single model from two gestures associated to two vocalizations, each being a variation within the same 'class' of gestures and sound[14]. We aimed to investigate if HMR can generate an intermediate sound from an

14 The question of how to define such classes of *consistent* relationship between gestures and vocal sounds remains open, as it relates both to our perception of invariants in movement and sound, and to the computational representation of these modalities.

intermediate *variation* of the gesture, without further specification of the variation itself. We plan to further investigate spectral envelope representations of both the harmonic and residual parts of the signal to reach a more consistent parametrization of the sound synthesis.

Bibliography

J. Aggarwal and Q. Cai, "Human motion analysis: a review," in *Proceedings IEEE Nonrigid and Articulated Motion Workshop*, vol. 73, no. 3. IEEE, 1997, pp. 90–102. (Cited on page 15.)

G. Ananthakrishnan, D. Neiberg, and O. Engwall, "In search of Non-uniqueness in the Acoustic-to-Articulatory Mapping," in *Proceedings of Interspeech*, 2009, pp. 2799–2802. (Cited on page 27.)

F. Anderson and W. F. Bischof, "Learning and performance with gesture guides," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13. Paris, France: ACM, 2013, pp. 1109–1118. (Cited on page 146.)

M. Anderson, "Embodied Cognition: A field guide," *Artificial Intelligence*, vol. 149, no. 1, pp. 91–130, Sep. 2003. (Cited on page 19.)

C. Appert and S. Zhai, "Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09. Boston, USA: ACM, 2009, pp. 2289–2298. (Cited on page 146.)

D. Arfib, J. M. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Organised Sound*, vol. 7, no. 02, pp. 127–144, 2002. (Cited on page 8.)

B. Argall and A. Billard, "Learning from Demonstration and Correction via Multiple Modalities for a Humanoid Robot," in *Proceedings of the International Conference SKILLS*. Montpellier, France: EDP Sciences, 2011, pp. 1–4. (Cited on page 28.)

B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009. (Cited on pages 29, 34, 39, and 189.)

P. Arias, "Description et synthèse sonore dans le cadre de l'apprentissage mouvement-son par démonstration," Master's Thesis, Ircam—Université Pierre et Marie Curie, 2014. (Cited on page 205.)

T. Artières, S. Marukatat, and P. Gallinari, "Online Handwritten Shape Recognition Using Segmental Hidden Markov Models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 2, pp. 205–217, 2007. (Cited on page 69.)

M. Astrinaki, N. D'Alessandro, B. Picart, T. Drugman, and T. Dutoit, "Reactive and continuous control of HMM-based speech synthesis," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, 2012, pp. 252–257. (Cited on page 25.)

M. Astrinaki, N. D'Alessandro, and T. Dutoit, "MAGE – A Platform for Tangible Speech Synthesis," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, G. Essl, B. Gillespie, M. Gurevich, and S. O'Modhrain, Eds. Ann Arbor, Michigan: University of Michigan, 2012. (Cited on page 25.)

J.-J. Aucouturier and F. Pachet, "Jamming with plunderphonics: Interactive concatenative synthesis of music," *Journal of New Music Research*, vol. 35, no. 1, pp. 35–50, 2006. (Cited on page 135.)

Y. Baram and A. Miller, "Auditory feedback control for improvement of gait in patients with Multiple Sclerosis," *Journal of the Neurological Sciences*, vol. 254, pp. 90–94, 2007. (Cited on page 147.)

I. Bartenieff, *Body movement: Coping with the environment*. Routledge, 1980. (Cited on pages 160 and 181.)

O. Bau and W. E. Mackay, "OctoPocus: A Dynamic Guide for Learning Gesture-based Command Sets," in *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08. ACM, 2008, pp. 37–46. (Cited on page 146.)

G. Beller, "The Synekine Project," in *Proceedings of the International Workshop on Movement and Computing*, MOCO'14. Paris, France: ACM, 2014, pp. 66–69. (Cited on page 166.)

R. Bencina, "The metasurface: applying natural neighbour interpolation to two-to-many mapping," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'05, Vancouver, Canada, 2005, pp. 101–104. (Cited on page 10.)

R. Bencina, D. Wilde, and S. Langley, "Gesture-sound experiments: Process and mappings," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME'08, Genova, Italy, 2008, pp. 197–202. (Cited on page 162.)

Y. Bengio, "Input-output HMMs for sequence processing," *Neural Networks, IEEE Transactions on*, vol. 7, no. 5, pp. 1231–1949, 1996. (Cited on page 27.)

P. A. Best, F. Levy, J. P. Fried, and F. Leventhal, "Dance and Other Expressive Art Therapies: When Words Are Not Enough," *Dance Research: The Journal of the Society for Dance Research*, vol. 16, no. 1, p. 87, Jan. 1998. (Cited on page 160.)

F. Bettens and T. Todoroff, "Real-time dtw-based gesture recognition external object for max/msp and puredata," *Proceedings of the SMC 2009 Conference*, vol. 9, no. July, pp. 30–35, 2009. (Cited on page 12.)

F. Bevilacqua, R. Muller, and N. Schnell, "MnM: a Max/MSP mapping toolbox," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'05, Vancouver, Canada, 2005. (Cited on pages 10, 11, and 13.)

F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," *Gesture in Embodied Communication and Human-Computer Interaction*, pp. 73–84, 2010. (Cited on pages 2, 13, 16, 55, and 86.)

F. Bevilacqua, N. Schnell, N. Rasamimanana, B. Zamborlin, and F. Guédy, "Online Gesture Analysis and Control of Audio Processing," in *Musical Robots and Interactive Multimodal Systems*. Springer, 2011, pp. 127–142. (Cited on pages 13, 14, 38, 63, 64, 77, and 86.)

A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Handbook of robotics*, vol. 1, 2008. (Cited on page 1.)

J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep., 1998. (Cited on page 45.)

——, "What HMMs Can Do," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 869—-891, 2006. (Cited on page 55.)

J. Bloit and X. Rodet, "Short-time Viterbi for online HMM decoding : Evaluation on a real-time phone recognition task," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. Ieee, 2008, pp. 2121–2124. (Cited on page 57.)

J. Bloit, N. Rasamimanana, and F. Bevilacqua, "Modeling and segmentation of audio descriptor profiles with segmental models," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1507–1513, Sep. 2010. (Cited on pages 15 and 69.)

I. Bowler, A. Purvis, P. Manning, and N. Bailey, "On mapping n articulation onto m synthesiser-control parameters," in *Proceedings of the International Computer Music Conference*, 1990, pp. 181–184. (Cited on page 10.)

M. Brand, "Voice puppetry," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 21–28. (Cited on pages 27 and 110.)

M. Brand and A. Hertzmann, "Style machines," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pp. 183–192, 2000. (Cited on page 28.)

B. Bruegge, C. Teschner, and P. Lachenmaier, "Pinocchio: conducting a virtual symphony orchestra," *Proceedings of the international conference on Advances in computer entertainment technology*, pp. 294–295, 2007. (Cited on page 12.)

C. Busso, Z. Deng, U. Neumann, and S. Narayanan, "Natural head motion synthesis driven by acoustic prosodic features," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 283–290, Jul. 2005. (Cited on page 27.)

W. Buxton, *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2010. (Cited on page 162.)

C. Cadoz, "Instrumental gesture and musical composition," in *Proceedings of the International computer music conference*, Cologne, Germany, 1988. (Cited on page 8.)

C. Cadoz, A. Luciani, and J.-L. Florens, "CORDIS-ANIMA: a Modeling and simulation system for sound and image synthesis: the general formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993. (Cited on page 9.)

S. Calinon, "Continuous extraction of task constraints in a robot programming by demonstration framework," PhD Dissertation, École Polytechnique Fédéral de Lausanne, 2007. (Cited on pages 104 and 191.)

S. Calinon, A. Pistillo, and D. G. Caldwell, "Encoding the time and space constraints of a task in explicit-duration Hidden Markov Model," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3413–3418, Sep. 2011. (Cited on pages 30, 55, and 124.)

S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing, and Generalizing a Task in a Humanoid Robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, Apr. 2007. (Cited on pages 29 and 124.)

S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 44–54, 2010. (Cited on pages 29, 112, and 124.)

S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, Nov. 2012, pp. 323–329. (Cited on page 29.)

S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 3339–3344. (Cited on page 30.)

A. Camurri, S. Hashimoto, and M. Ricchetti, "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems," *Computer Music Journal*, vol. 24, no. 1, pp. 57–69, 2000. (Cited on page 11.)

A. Camurri, "Movement and Gesture in Intelligent interactive Music Systems," *Trends in Gestural Control of Music*, pp. 95–110, 2000. (Cited on page 11.)

A. Camurri, M. Ricchetti, and R. Trocca, "EyesWeb-toward gesture and affect recognition in dance/music interactive systems," *Proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 1, no. 1, pp. 643–648, 2000. (Cited on page 180.)

A. Camurri, B. Mazzarino, and M. Ricchetti, "Multimodal analysis of expressive gesture in music and dance performances," *Gesture-Based Communication in Human-Computer Interaction*, pp. 20–39, 2004. (Cited on page 11.)

B. Caramiaux, "Studies on the Relationship between Gesture and Sound in Musical Performance," Ph.D. dissertation, Université Pierre et Marie Curie (Paris 6) and UMR STMS IRCAM CNRS UPMC, 2012. (Cited on pages 14 and 33.)

B. Caramiaux, F. Bevilacqua, T. Bianco, N. Schnell, O. Houix, and P. Susini, "The Role of Sound Source Perception in Gestural Sound Description," *ACM Transactions on Applied Perception*, vol. 11, no. 1, pp. 1–19, Apr. 2014. (Cited on pages 20, 33, and 134.)

B. Caramiaux, "Motion Modeling for Expressive Interaction - A Design Proposal using Bayesian Adaptive Systems," in *Proceedings of the International Workshop on Movement and Computing*, MOCO'14. Paris, France: ACM, 2014, pp. 76–81. (Cited on page 2.)

B. Caramiaux and A. Tanaka, "Machine Learning of Musical Gestures," in *proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2013)*, Seoul, South Korea, 2013. (Cited on page 33.)

B. Caramiaux, F. Bevilacqua, and N. Schnell, "Analysing Gesture and Sound Similarities with a HMM-based Divergence Measure," in *Proceedings of the Sound and Music Computing Conference, SMC*, 2010. (Cited on page 20.)

——, "Towards a gesture-sound cross-modal analysis," *Gesture in Embodied Communication and Human-Computer Interaction*, vol. 5934, pp. 158–170, 2010. (Cited on page 134.)

——, "Sound Selection by Gestures," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt, Eds., Oslo, Norway, 2011, pp. 329–330. (Cited on page 69.)

B. Caramiaux, M. M. Wanderley, and F. Bevilacqua, "Segmenting and Parsing Instrumentalist's Gestures," *Journal of New Music Research*, vol. 41, no. 1, pp. 1–27, 2012. (Cited on page 15.)

B. Caramiaux, J. Françoise, N. Schnell, and F. Bevilacqua, "Mapping Through Listening," *Computer Music Journal*, vol. 38, no. 3, pp. 34–48, 2014. (Cited on pages 21, 33, and 37.)

B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua, "Adaptive Gesture Recognition with Variation Estimation for Interactive Systems,"

ACM Trans. Interact. Intell. Syst., vol. 4, no. 4, pp. 18:1—-18:34, 2014. (Cited on pages 13, 16, and 33.)

M. Cartwright and B. Pardo, "SynthAssist: Querying an Audio Synthesizer by Vocal Imitation," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, B. Caramiaux, K. Tahiroglu, R. Fiebrink, and A. Tanaka, Eds. London, United Kingdom: Goldsmiths, University of London, 2014, pp. 363–366. (Cited on page 161.)

N. Castagné and C. Cadoz, "GENESIS: a Friendly Musician-Oriented Environment for Mass-Interaction Physical Modeling," in *Proceedings of the International Computer Music Conference*, Gothenburg, Sweden, 2002, pp. 330–337. (Cited on page 9.)

N. Castagne, C. Cadoz, J.-L. Florens, and A. Luciani, "Haptics in computer music: a paradigm shift," *EuroHaptics*, 2004. (Cited on page 8.)

R. E. Causse, J. Bensoam, and N. Ellis, "Modalys, a physical modeling synthesizer: More than twenty years of researches, developments, and musical uses," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, 2011. (Cited on page 117.)

T. Chen, "Audiovisual speech processing," *Signal Processing Magazine, IEEE*, vol. 18, no. 1, pp. 9–21, Jan. 2001. (Cited on pages 27 and 111.)

K. Choi, Y. Luo, and J.-n. Hwang, "Hidden Markov model inversion for audio-to-visual conversion in an MPEG-4 facial animation system," *The Journal of VLSI Signal Processing*, vol. 29, no. 1, pp. 51–61, 2001. (Cited on pages 27, 109, and 111.)

A. Cont, T. Coduys, and C. Henry, "Real-time Gesture Mapping in Pd Environment using Neural Networks," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'04, Hamamatsu, Japan, 2004, pp. 39–42. (Cited on page 15.)

K. Dautenhahn and C. L. Nehaniv, *The correspondence problem*. MIT Press, 2002. (Cited on page 39.)

M. Demoucron, "On the control of virtual violins - Physical modelling and control of bowed string instruments," PhD dissertation, UPMC (Paris) and KTH (Stockholm), 2008. (Cited on page 8.)

L. Deng and X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, May 2013. (Cited on page 22.)

P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, pp. 4–7, 1993. (Cited on page 207.)

J. Dines, J. Yamagishi, and S. King, "Measuring the Gap Between HMM-Based ASR and TTS," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1046–1058, Dec. 2010. (Cited on page 23.)

Y. Ding, "Data-Driven Expressive Animation Model of Speech and Laughter for an Embodied Conversational Agent," PhD Dissertation, TELECOM ParisTech, 2014. (Cited on pages 27 and 191.)

Y. Ding, M. Radenen, T. Artieres, and C. Pelachaud, "Speech-driven eyebrow motion synthesis with contextual Markovian models," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 3756–3760. (Cited on page 27.)

P. Dourish, *Where the action is*. The MIT Press, 2004. (Cited on page 20.)

G. Dubus and R. Bresin, "A systematic review of mapping strategies for the sonification of physical quantities." *PloS one*, vol. 8, no. 12, p. e82491, Jan. 2013. (Cited on page 147.)

T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artificial Intelligence*, vol. 173, no. 7-8, pp. 830–856, May 2009. (Cited on page 15.)

A. Effenberg, U. Fehse, and A. Weber, "Movement Sonification: Audiovisual benefits on motor learning," *BIO Web of Conferences*, vol. 1, Dec. 2011. (Cited on page 147.)

S. Eickeler, A. Kosmala, and G. Rigoll, "Hidden markov model based continuous online gesture recognition," *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, pp. 1206–1208, 1998. (Cited on page 52.)

I. Ekman and M. Rinott, "Using vocal sketching for designing sonic interactions," in *Proceedings of the 8th ACM conference on designing interactive systems*. ACM, 2010, pp. 123–131. (Cited on page 162.)

P. Esling and C. Agon, "Multiobjective time series matching for audio classification and retrieval," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, pp. 2057–2072, 2013. (Cited on page 161.)

J. A. Fails and D. R. Olsen, "Interactive machine learning," in *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI'03, 2003, pp. 39–45. (Cited on page 16.)

S. Fasciani and L. Wyse, "A Voice Interface for Sound Generators: adaptive and automatic mapping of gestures to sound," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, G. Essl, B. Gillespie, M. Gurevich, and S. O'Modhrain, Eds. Ann Arbor, Michigan: University of Michigan, 2012. (Cited on page 163.)

S. Fdili Alaoui, B. Caramiaux, M. Serrano, and F. Bevilacqua, "Movement Qualities as Interaction Modality," in *ACM Designing Interactive Systems*, DIS'12, Newcastle, UK, 2012. (Cited on pages 179 and 180.)

S. Fdili Alaoui, F. Bevilacqua, B. B. Pascual, and C. Jacquemin, "Dance interaction with physical model visuals based on movement qualities," *International Journal of Arts and Technology*, vol. 6, no. 4, pp. 357–387, 2013. (Cited on page 180.)

S. Fels and G. Hinton, "Glove-talkII: A neural network interface between a data-glove and a speech synthesizer," *Neural Networks, IEEE Transactions on*, vol. 4, no. 1, pp. 2–8, 1993. (Cited on page 15.)

S. Ferguson, E. Schubert, and C. Stevens, "Dynamic dance warping: Using dynamic time warping to compare dance movement performed under different conditions," in *Proceedings of the International Workshop on Movement and Computing*, MOCO'14. Paris, France: ACM, 2014, pp. 94–99. (Cited on page 85.)

R. Fiebrink, G. Wang, and P. Cook, "Don't forget the laptop: Using native input capabilities for expressive musical control," in *New Interfaces for Musical Expression (NIME 2007)*, 2007. (Cited on page 11.)

R. Fiebrink, P. R. Cook, and D. Trueman, "Play-along mapping of musical controllers," in *In Proceedings of the International Computer Music Conference*, 2009. (Cited on page 34.)

——, "Human model evaluation in interactive supervised learning," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'11. Vancouver, BC, Canada: ACM, 2011, p. 147. (Cited on pages 18, 34, and 47.)

R. A. Fiebrink, "Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance," Ph.D. dissertation, Faculty of Princeton University, 2011. (Cited on pages 1, 12, 15, 16, 18, and 34.)

S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998. (Cited on pages 69, 70, and 72.)

J. Fogarty, D. Tan, A. Kapoor, and S. Winder, "CueFlik: interactive concept learning in image search," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems*, CHI '08, 2008, p. 29. (Cited on page 17.)

J. Françoise, "Realtime Segmentation and Recognition of Gestures using Hierarchical Markov Models," Master's Thesis, Université Pierre et Marie Curie, Ircam, 2011. (Cited on pages 15, 69, 70, and 72.)

J. Françoise, B. Caramiaux, and F. Bevilacqua, "A Hierarchical Approach for the Design of Gesture-to-Sound Mappings," in *Proceedings of the 9th Sound and Music Computing Conference*, Copenhagen, Denmark, 2012, pp. 233–240. (Cited on pages 75, 78, and 198.)

J. Françoise, N. Schnell, and F. Bevilacqua, "Gesture-based control of physical modeling sound synthesis," in *Proceedings of the 21st ACM international conference on Multimedia (MM'13)*. Barcelona, Spain: ACM Press, 2013, pp. 447–448. (Cited on page 117.)

——, "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, 2013, pp. 705–708. (Cited on pages 117 and 198.)

J. Françoise, S. Fdili Alaoui, T. Schiphorst, and F. Bevilacqua, "Vocalizing Dance Movement for Interactive Sonification of Laban Effort Factors," in *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14. Vancouver, Canada: ACM, 2014, pp. 1079–1082. (Cited on page 179.)

J. Françoise, N. Schnell, and F. Bevilacqua, "MaD: Mapping by Demonstration for Continuous Sonification," in *ACM SIGGRAPH 2014 Emerging Technologies*, SIGGRAPH '14. Vancouver, Canada: ACM, 2014, pp. 16:1—-16:1. (Cited on page 166.)

J. Françoise, N. Schnell, R. Borghesi, and F. Bevilacqua, "Probabilistic Models for Designing Motion and Sound Relationships," in *Proceedings of the 2014 International Conference on New Interfaces for Musical Expression*, NIME'14, London, UK, 2014, pp. 287–292. (Cited on pages 48 and 196.)

K. Franinović and S. Serafin, *Sonic Interaction Design*. Mit Press, 2013. (Cited on page 7.)

S. Fu, R. Gutierrez-Osuna, A. Esposito, P. Kakumanu, and O. Garcia, "Audio/visual mapping with cross-modal hidden Markov models," *Multimedia, IEEE Transactions on*, vol. 7, no. 2, pp. 243–252, Apr. 2005. (Cited on pages 2, 27, 109, and 111.)

M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2007. (Cited on page 22.)

W. Gaver, "Auditory Icons: Using Sound in Computer Interfaces," *Human-Computer Interaction*, vol. 2, no. 2, pp. 167–177, Jun. 1986. (Cited on page 146.)

W. W. Gaver, "How Do We Hear in the World? Explorations in Ecological Acoustics," *Ecological Psychology*, vol. 5, no. 4, pp. 285–313, Dec. 1993. (Cited on page 134.)

S. Gelineck and N. Böttcher, "An educational tool for fast and easy mapping of input devices to musical parameters," in *Audio Mostly, conference on interaction with sound*, 2012. (Cited on page 11.)

Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems*, 1994. (Cited on pages 26, 104, and 106.)

N. Gillian and R. Knapp, "A Machine Learning Toolbox For Musician Computer Interaction," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'11, Oslo, Norway, 2011. (Cited on pages 12 and 16.)

N. Gillian and J. A. Paradiso, "Digito: A Fine-Grain Gesturally Controlled Virtual Musical Instrument," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'12, Ann Arbor, Michigan, 2012. (Cited on page 12.)

——, "The Gesture Recognition Toolkit," *Journal of Machine Learning Research*, vol. 15, pp. 3483–3487, 2014. (Cited on pages 12 and 16.)

N. Gillian, B. Knapp, and S. O'Modhrain, "Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'11. Oslo, Norway: Oslo, Norway, 2011, pp. 337–342. (Cited on page 85.)

N. E. Gillian, "Gesture Recognition for Musician Computer Interaction," PhD dissertation, Faculty of Arts, Humanities and Social Sciences, 2011. (Cited on page 12.)

R. I. Godøy, "Motor-mimetic music cognition," *Leonardo*, vol. 36, no. 4, pp. 317–319, 2003. (Cited on page 20.)

R. I. Godøy, E. Haga, and A. Jensenius, "Playing " Air Instruments ": Mimicry of Sound-producing Gestures by Novices and Experts," *Gesture in Human-Computer Interaction and Simulation*, pp. 256–267, 2006. (Cited on pages 20 and 134.)

——, "Exploring music-related gestures by sound-tracing.-a preliminary study," in *2nd ConGAS International Symposium on Gesture Interfaces for Multimedia Systems*, 2006, pp. 9–10. (Cited on pages 20 and 134.)

R. I. Godøy, A. R. Jensenius, and K. Nymoen, "Chunking in music by coarticulation," *Acta Acustica united with Acustica*, vol. 96, no. 4, pp. 690–700, 2010. (Cited on page 14.)

R. Goudard, H. Genevois, E. Ghomi, and B. Doval, "Dynamic Intermediate Models for audiographic synthesis," in *Proceedings of the Sound and Music Computing Conference*, SMC'11, 2011. (Cited on page 9.)

C. Goudeseune, "Interpolated mappings for musical instruments," *Organised Sound*, vol. 7, no. 2, pp. 85–96, 2002. (Cited on page 10.)

T. Grossman, P. Dragicevic, and R. Balakrishnan, "Strategies for accelerating on-line learning of hotkeys," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI'07. ACM, 2007, p. 1591. (Cited on page 146.)

G. Guerra-Filho and Y. Aloimonos, "A Language for Human Action," *Computer*, vol. 40, no. 5, pp. 42–51, 2007. (Cited on page 15.)

B. Hartmann, L. Abdulla, M. Mittal, and S. R. Klemmer, "Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI'07. New York, New York, USA: ACM Press, 2007, p. 145. (Cited on page 17.)

C. Henry, "Physical modeling for pure data (PMPD) and real time interaction with an audio synthesis," in *Proceedings of the 2004 Sound and Music Computing Conference*, F. Paris, Ed., P, 2004. (Cited on page 9.)

T. Hermann, J. Neuhoff, and A. Hunt, *The Sonification Handbook*. Logos Verlag, Berlin, Germany, 2011. (Cited on page 7.)

G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012. (Cited on pages 22 and 23.)

G. Hofer, "Speech-driven animation using multi-modal hidden Markov models," PhD Dissertation, University of Edimburgh, 2009. (Cited on page 109.)

D. M. Howard and S. Rimell, "Real-Time Gesture-Controlled Physical Modelling Music Synthesis with Tactile Feedback," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 1001–1006, 2004. (Cited on page 8.)

T. Hueber and P. Badin, "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. (Cited on pages 3 and 26.)

A. Hunt and R. Kirk, "Mapping Strategies for Musical Performance," *Trends in Gestural Control of Music*, pp. 231–258, 2000. (Cited on page 8.)

A. Hunt and M. M. Wanderley, "Mapping performer parameters to synthesis engines," *Organised Sound*, vol. 7, no. 02, pp. 97–108, 2002. (Cited on page 8.)

A. Hunt, M. M. Wanderley, and R. Kirk, "Towards a Model for Instrumental Mapping in Expert Musical Interaction," in *Proceedings of the 2000 International Computer Music Conference*, 2000, pp. 209–212. (Cited on page 8.)

A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors." *Neural computation*, vol. 25, no. 2, pp. 328–73, Feb. 2013. (Cited on page 29.)

J. Janer, "Singing-driven interfaces for sound synthesizers," PhD Dissertation, Universitat Pompeu Fabra, 2009. (Cited on page 162.)

A. Johnston, "Interfaces for musical expression based on simulated physical models," Ph.D. dissertation, University of Technology, Sydney, 2009. (Cited on page 9.)

A. Johnston, B. Marks, and L. Candy, "Sound controlled musical instruments based on physical models," in *Proceedings of the 2007*

International Computer Music Conference, 2007, pp. 232–239. (Cited on page 9.)

A. Johnston, L. Candy, and E. Edmonds, "Designing and evaluating virtual musical instruments: facilitating conversational user interaction," *Design Studies*, vol. 29, no. 6, pp. 556–571, Nov. 2008. (Cited on page 10.)

S. Jorda, M. Kaltenbrunner, G. Geiger, and R. Bencina, "The reactable*," *Proceedings of the international computer music conference (ICMC 2005), Barcelona, Spain*, pp. 579—-582, 2005. (Cited on page 12.)

S. Jordà, "Digital Lutherie: Crafting musical computers for new musics performance and improvisation," PhD Dissertation, Universitat Pompeu Fabra, 2005. (Cited on page 14.)

——, "On Stage: the Reactable and other Musical Tangibles go Real," *International Journal of Arts and Technology*, vol. 1, pp. 268–287, 2008. (Cited on pages 12 and 14.)

B.-H. Juang and L. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 9, pp. 1639–1641, 1990. (Cited on page 59.)

H. Kang, "Recognition-based gesture spotting in video games," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1701–1714, Nov. 2004. (Cited on page 52.)

H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, Apr. 1999. (Cited on page 165.)

G. Kellum and A. Crevoisier, "A Flexible Mapping Editor for Multi-touch Musical Instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2009, pp. 242–245. (Cited on page 11.)

A. Kendon, "Current issues in the study of gesture," *The biological foundations of gestures: Motor and semiotic aspects*, vol. 1, pp. 23–47, 1986. (Cited on pages 69 and 76.)

E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, May 2004. (Cited on page 85.)

C. Kiefer, "Musical Instrument Mapping Design with Echo State Networks," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME'14. London, United Kingdom: Goldsmiths, University of London, 2014, pp. 293–298. (Cited on page 15.)

D. Kim, J. Song, and D. Kim, "Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs," *Pattern Recognition*, vol. 40, no. 11, pp. 3012–3026, Nov. 2007. (Cited on page 52.)

D. Kirsh, "Embodied cognition and the magical future of interaction design," *ACM Transactions on Computer-Human Interaction*, vol. 20, no. 111, pp. 3:1–3:30, 2013. (Cited on page 20.)

D. Kirsh, D. Muntanyola, and R. Jao, "Choreographic methods for creating novel, high quality dance," in *5th International workshop on Design and Semantics of Form and Movement.*, 2009. (Cited on page 161.)

W. Knox, "Learning from Human-Generated Reward," PhD Dissertation, University of Texas, Austin, 2012. (Cited on page 16.)

E. Kohler, C. Keysers, M. A. Umiltà, L. Fogassi, V. Gallese, and G. Rizzolatti, "Hearing sounds, understanding actions: action representation in mirror neurons." *Science*, vol. 297, pp. 846–848, 2002. (Cited on pages 20 and 134.)

P. Kolesnik, "Recognition, analysis and performance with expressive conducting gestures," in *Proceedings of the International Computer Music Conference*, ICMC'04, 2004. (Cited on page 12.)

P. Kolesnik and M. M. Wanderley, "Implementation of the Discrete Hidden Markov Model in Max/MSP Environment," in *FLAIRS Conference*, 2005, pp. 68–73. (Cited on page 12.)

D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. Cambridge, Massachusetts: MIT press, 2009. (Cited on page 2.)

T. Kulesza, "Personalizing Machine Learning Systems with Explanatory Debugging," PhD Dissertation, Oregon State University, 2014. (Cited on page 16.)

G. Kurtenbach and W. Buxton, "User learning and performance with marking menus," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1994, pp. 258–264. (Cited on page 146.)

R. Laban and F. C. Lawrence, *Effort*. London: MacDonald and Evans., 1947. (Cited on page 180.)

C. P. Laguna and R. Fiebrink, "Improving data-driven design and exploration of digital musical instruments," in *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*, CHI EA '14. New York, New York, USA: ACM Press, 2014, pp. 2575–2580. (Cited on page 19.)

G. Lakoff and M. Johnson, *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*, Collection of Jamie and Michael Kassler. Basic Books, 1999. (Cited on page 19.)

E. W. Large, "On synchronizing movements to music," *Human Movement Science*, vol. 19, no. 4, pp. 527–566, Oct. 2000. (Cited on page 134.)

J. J. LaViola, "3D Gestural Interaction: The State of the Field," *International Scholarly Research Notices*, vol. 2013, 2013. (Cited on page 1.)

E. Lee and T. Nakra, "You're the conductor: a realistic interactive conducting system for children," in *Proceedings of International Conference on New Interfaces for Musical Expression*, NIME'04, 2004, pp. 68–73. (Cited on page 12.)

H.-k. Lee and J. H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 10, pp. 961–973, 1999. (Cited on page 55.)

M. Lee, A. Freed, and D. Wessel, "Neural networks for simultaneous classification and parameter estimation in musical instrument control," *Adaptive and Learning Systems*, vol. 1706, pp. 244–55, 1992. (Cited on page 15.)

C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995. (Cited on pages 25 and 191.)

G. Lemaitre and D. Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds." *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–73, Feb. 2014. (Cited on page 161.)

G. Lemaitre, O. Houix, N. Misdariis, and P. Susini, "Listener expertise and sound identification influence the categorization of environmental sounds." *Journal of experimental psychology. Applied*, vol. 16, no. 1, pp. 16–32, Mar. 2010. (Cited on page 134.)

M. Leman, *Embodied Music Cognition and mediation technology*. The MIT Press, 2008. (Cited on pages 20 and 190.)

Y. Li and H.-y. Shum, "Learning dynamic audio-visual mapping with input-output Hidden Markov models," *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 542–549, Jun. 2006. (Cited on page 27.)

O. Liam, F. Dermot, and F. Boland, "Introducing CrossMapper: Another Tool for Mapping Musical Control Parameters," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME'12. Ann Arbor, Michigan: University of Michigan, 2012. (Cited on page 11.)

J. D. Loehr and C. Palmer, "Cognitive and biomechanical influences in pianists' finger tapping." *Experimental Brain Research*, vol. 178, no. 4, pp. 518–28, Apr. 2007. (Cited on page 14.)

M. Logan, "Dance in the schools: A personal account," *Theory Into Practice*, vol. 23, no. 4, pp. 200–302, 1984. (Cited on page 160.)

H. Lü and Y. Li, "Gesture coder: a tool for programming multi-touch gestures by demonstration," in *Proceedings of the ACM annual conference on Human Factors in Computing Systems*, CHI '12. Austin, Texas, USA: ACM Press, 2012, p. 2875. (Cited on page 17.)

J. Malloch, S. Sinclair, and M. M. Wanderley, "A Network-Based Framework for Collaborative Development and Performance of Digital Musical Instruments," in *Computer Music Modeling and Retrieval. Sense of Sounds*. Springer, 2008, pp. 401–425. (Cited on page 11.)

D. S. Maranan, S. Fdili Alaoui, T. Schiphorst, P. Pasquier, P. Subyen, and L. Bartram, "Designing for movement: evaluating computational models using LMA effort qualities," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, CHI '14. Toronto, Canada: ACM Press, 2014, pp. 991–1000. (Cited on page 180.)

T. Marrin and R. Picard, "The 'Conductor's Jacket': A Device for Recording Expressive Musical Gestures," in *Proceedings of the International Computer Music Conference*, no. 470, 1998, pp. 215–219. (Cited on page 12.)

M. V. Mathews, "The radio baton and conductor program, or: Pitch, the most important and least expressive part of music," *Computer Music Journal*, pp. 37–46, 1991. (Cited on page 7.)

G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004. (Cited on page 44.)

H. M. Mentis and C. Johansson, "Seeing movement qualities," in *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*, CHI'13. Paris, France: ACM, 2013, pp. 3375–3384. (Cited on page 180.)

D. J. Merrill and J. A. Paradiso, "Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. Portland, OR, USA.: ACM Press, 2005, p. 2152. (Cited on page 17.)

G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information." *Psychological Review*, vol. 101, no. 2, pp. 343–352, Apr. 1956. (Cited on page 14.)

E. Miranda and M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006. (Cited on page 7.)

S. Mitra, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C:*, vol. 37, no. 3, pp. 311–324, 2007. (Cited on page 55.)

P. Modler, "Neural Networks for Mapping Hand Gestures to Sound Synthesis parameters," *Trends in Gestural Control of Music*, pp. 301–314, 2000. (Cited on pages 12 and 15.)

P. Modler, T. Myatt, and M. Saup, "An Experimental Set of Hand Gestures for Expressive Control of Musical Parameters in Realtime," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME'03. McGill University, 2003, pp. 146–150. (Cited on page 12.)

A. Momeni and C. Henry, "Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis," *Computer Music Journal*, vol. 30, no. 1, pp. 49–66, 2006. (Cited on page 9.)

K. P. Murphy and M. A. Paskin, "Linear Time Inference in Hierarchical HMMs," in *Advances in Neural Information Processing Systems 14, Proceedings of the 2001 Conference*. MIT press, 2001, pp. 833–840. (Cited on pages 70, 72, and 74.)

K. P. Murphy, "Hidden semi-Markov models," Tech. Rep. November, 2002. (Cited on page 69.)

——, "Dynamic bayesian networks: representation, inference and learning," PhD Thesis, University of California, Berkeley, 2002. (Cited on page 72.)

——, "Gaussians," Tech. Rep., 2006. (Cited on page 51.)

——, *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: MIT press, 2012. (Cited on pages 2, 45, 48, 55, 57, and 59.)

M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, "Memorability of Pre-designed and User-defined Gesture Sets," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13. ACM, 2013, pp. 1099–1108. (Cited on page 146.)

M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay, "Mid-air pan-and-zoom on wall-sized displays," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, ACM. Vancouver, BC, Canada: ACM Press, 2011, p. 177. (Cited on page 146.)

U. Oh and L. Findlater, "The challenges and potential of end-user gesture customization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. New York, New York, USA: ACM Press, 2013, p. 1129. (Cited on page 146.)

U. Oh, S. K. Kane, and L. Findlater, "Follow that sound: using sonification and corrective verbal feedback to teach touchscreen gestures," in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '13*. New York, New York, USA: ACM Press, 2013, pp. 1–8. (Cited on page 147.)

J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness." *The Behavioral and brain sciences*, vol. 24, pp. 939–973; discussion 973–1031, 2001. (Cited on page 19.)

D. Ormoneit and V. Tresp, “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, 1996, pp. 542–548. (Cited on page 48.)

M. Ostendorf, V. Digalakis, and O. Kimball, “From HMM’s to segment models: A unified view of stochastic modeling for speech recognition,” *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 5, pp. 360–378, 1996. (Cited on pages 15 and 69.)

S. Park and J. Aggarwal, “A hierarchical bayesian network for event recognition of human actions and interactions,” *Multimedia Systems*, vol. 10, no. 2, pp. 164–179, 2004. (Cited on page 15.)

K. Patel, N. Bancroft, S. M. Drucker, J. Fogarty, A. J. Ko, and J. Landay, “Gestalt: integrated support for implementation and analysis in machine learning,” in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010, pp. 37–46. (Cited on page 17.)

V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual interpretation of hand gestures for human-computer interaction: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 677–695, 1997. (Cited on page 69.)

D. Pirrò and G. Eckel, “Physical modelling enabling enaction: an example,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt, Eds., Oslo, Norway, 2011, pp. 461–464. (Cited on page 10.)

L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. (Cited on pages 55, 57, 58, and 59.)

A. Rahimi, “An Erratum for "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”,” Tech. Rep., 2000. [Online]. Available: http://alumni.media.mit.edu/~rahimi/rabiner/rabiner-errata/ (Cited on pages 55 and 59.)

R. Rao, T. Chen, and R. Mersereau, “Audio-to-visual conversion for multimedia communication,” *IEEE Transactions on Industrial Electronics*, vol. 45, no. 1, pp. 15–22, 1998. (Cited on page 27.)

N. Rasamimanana and J. Bloit, “Reconnaissance de timbres en temps-réel et Applications,” p. Private Communication, 2011. (Cited on page 49.)

N. Rasamimanana, F. Bevilacqua, N. Schnell, E. Fléty, and B. Zamborlin, “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, 2011, pp. 9–12. (Cited on pages 78, 84, 117, 200, and 203.)

T. Ravet, J. Tilmanne, and N. D'Alessandro, "Hidden Markov Model Based Real-Time Motion Recognition and Following," in *Proceedings of the International Workshop on Movement and Computing*, MOCO'14. Paris, France: ACM, 2014, pp. 82–87. (Cited on page 57.)

K. Richmond, "A Trajectory Mixture Density Network for the Acoustic-Articulatory Inversion Mapping," in *Proceedings of Interspeech*, 2006, pp. 577–580. (Cited on page 26.)

J. V. G. Robertson, T. Hoellinger, P. v. Lindberg, D. Bensmail, S. Hanneton, and A. Roby-Brami, "Effect of auditory feedback differs according to side of hemiparesis: a comparative pilot study." *Journal of neuroengineering and rehabilitation*, vol. 6, p. 45, Jan. 2009. (Cited on page 147.)

D. Rocchesso, G. Lemaitre, P. Susini, S. Ternström, and P. Boussard, "Sketching Sound with Voice and Gesture," *interactions*, vol. 22, no. 1, pp. 38–41, 2015. (Cited on pages 162, 167, 186, and 192.)

X. Rodet and P. Depalle, "Spectral envelopes and inverse FFT synthesis," 1992. (Cited on page 207.)

J. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle, "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance," in *Proceedings of the AIMI International Workshop.*, 1997, pp. 68–73. (Cited on pages 7, 8, and 20.)

M. J. Russell, "A segmental HMM for speech pattern modelling," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93*, vol. 2. IEEE, 1993, pp. 499–502. (Cited on page 15.)

T. Sanger, "Bayesian filtering of myoelectric signals," *Journal of neurophysiology*, vol. 97, no. 2, pp. 1839–1845, 2007. (Cited on page 205.)

M. Savary, D. Schwarz, D. Pellerin, F. Massin, C. Jacquemin, and R. Cahen, "Dirty tangible interfaces: expressive control of computers with true grit," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems on*, CHI EA '13. Paris, France: ACM Press, 2013, p. 2991. (Cited on page 135.)

H. Sawada and S. Hashimoto, "Gesture recognition using an acceleration sensor and its application to musical performance control," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 80, no. 5, pp. 1520–6440, 1997. (Cited on page 12.)

S. Schaal, "Is imitation learning the route to humanoid robots," *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999. (Cited on pages 28 and 34.)

S. Schaal, S. Kotosaka, and D. Sternad, "Nonlinear dynamical systems as movement primitives," in *IEEE International Conference on Humanoid Robotics*, 2000, pp. 1–11. (Cited on page 29.)

S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation." *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 358, no. 1431, pp. 537–47, Mar. 2003. (Cited on pages 29 and 34.)

T. Schiphorst, "soft (n): Toward a Somaesthetics of Touch," in *Proceedings of the International Conference on Human Factors in Computing Systems*, CHI'09. Boston, USA: ACM, 2009, pp. 2427–2438. (Cited on page 180.)

N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "Mubu & friends - assembling tools for content based real-time interactive audio processing in max/msp," in *Proceedings of International Computer Music Conference*, Montreal, 2009. (Cited on pages 80, 138, and 199.)

D. Schwarz, "Data-driven concatenative sound synthesis," PhD Dissertation, Université Pierre et Marie Curie and Ircam, 2004. (Cited on page 135.)

——, "State of the art in sound texture synthesis," in *Proc. of the 14th Int. Conference on Digital Audio Effects*, DAFx-11, 2011, pp. 221–231. (Cited on page 135.)

——, "Corpus-based concatenative synthesis," *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 92–104, 2007. (Cited on pages 135 and 203.)

——, "The Sound Space as Musical Instrument: Playing Corpus-Based Concatenative Synthesis," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. (Cited on page 135.)

D. Schwarz and B. Caramiaux, "Interactive Sound Texture Synthesis Through Semi-Automatic User Annotations," in *Sound, Music, and Motion*, lecture no ed. Springer International Publishing, 2014, pp. 372–392. (Cited on page 135.)

D. Schwarz, G. Beller, and B. Verbrugghe, "Real-time corpus-based concatenative synthesis with catart," *Proc. of the Int. Conf. on*, no. September, pp. 1–7, 2006. (Cited on page 135.)

S. Sentürk, S. Lee, A. Sastry, A. Daruwalla, and G. Weinberg, "Crossole: A Gestural Interface for Composition, Improvisation and Performance using Kinect," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME'12, Ann Arbor, Michigan, 2012. (Cited on page 12.)

S. Serafin, M. Burtner, C. Nichols, and S. O'Modhrain, "Expressive controllers for bowed string physical models," in *Proceedings of the COST G-6 Conference on Digital Audio Effects*, Limerick, Ireland, 2001, pp. 6–9. (Cited on page 8.)

X. Serra, "A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition," PhD Dissertation, Stanford University, 1989. (Cited on page 207.)

B. Settles, "Active learning literature survey," University of Wisconsin, Madison, Tech. Rep., 2009. (Cited on page 16.)

R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review." *Psychonomic bulletin & review*, vol. 20, no. 1, pp. 21–53, Mar. 2013. (Cited on page 147.)

R. Sramek, "The on-line Viterbi algorithm," Master's Thesis, Comenius University, Bratislava, 2007. (Cited on page 57.)

D. Stowell, "Making music through real-time voice timbre analysis: machine learning and timbral control," PhD Dissertation, 2010. (Cited on page 163.)

H. G. Sung, "Gaussian Mixture Regression and Classification," PhD Dissertation, Rice University, Houston, TX, 2004. (Cited on pages 104 and 197.)

K. Tahiroglu and T. Ahmaniemi, "Vocal Sketching : a Prototype Tool for Designing Multimodal Interaction," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010, pp. 1–4. (Cited on page 162.)

J. Talbot, B. Lee, A. Kapoor, and D. S. Tan, "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, 2009, pp. 1283–1292. (Cited on page 17.)

P. Taylor, *Text-to-Speech Synthesis*, 1st ed. Cambridge University Press, 2009. (Cited on page 22.)

J. Tilmanne, "Data-driven Stylistic Humanlike Walk Synthesis," PhD Dissertation, University of Mons, 2013. (Cited on pages 28 and 55.)

J. Tilmanne, A. Moinet, and T. Dutoit, "Stylistic gait synthesis based on hidden Markov models," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, p. 72, 2012. (Cited on pages 2 and 28.)

T. Toda and K. Tokuda, "A Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 816–824, May 2007. (Cited on pages 25 and 190.)

T. Toda, A. W. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with gaussian mixture model." in *INTERSPEECH*, 2004. (Cited on pages 26 and 104.)

——, "Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007. (Cited on page 25.)

——, “Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model,” *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008. (Cited on pages 26, 104, and 106.)

K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “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, 2000, pp. 1315–1318 vol.3. (Cited on pages 24 and 111.)

K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech Synthesis Based on Hidden Markov Models,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013. (Cited on pages 2, 23, 24, and 55.)

S. Trail, M. Dean, T. Tavares, and G. Odowichuk, “Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect,” *New Interfaces for Musical Expression (NIME 2012)*, 2012. (Cited on page 12.)

P. A. Tremblay and D. Schwarz, “Surfing the Waves: Live Audio Mosaicing of an Electric Bass Performance as a Corpus Browsing Interface,” *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), Sydney, Australia*, no. June, pp. 15–18, 2010. (Cited on page 135.)

D. Van Nort, M. M. Wanderley, and P. Depalle, “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, 2004, pp. 87–91. (Cited on page 10.)

D. Van Nort and M. M. Wanderley, “The LoM Mapping Toolbox for Max/MSP/Jitter,” in *Proc. of the 2006 International Computer Music Conference (ICMC)*, 2006, pp. 397–400. (Cited on page 11.)

D. Van Nort, M. M. Wanderley, and P. Depalle, “Mapping Control Structures for Sound Synthesis: Functional and Topological Perspectives,” *Comput. Music J.*, vol. 38, no. 3, pp. 6–22, 2014. (Cited on pages 10 and 21.)

F. Vass-Rhee, “Dancing music: The intermodality of The Forsythe Company,” in *William Forsythe and the Practice of Choreography*, S. Spier, Ed., 2010, pp. 73–89. (Cited on page 161.)

——, “Auditory Turn: William Forsythe’s Vocal Choreography,” *Dance Chronicle*, vol. 33, no. 3, pp. 388–413, Nov. 2010. (Cited on pages 161 and 163.)

V. Verfaille, M. M. Wanderley, and P. Depalle, “Mapping strategies for gestural and adaptive control of digital audio effects,” *Journal of New Music Research*, vol. 35, no. 1, pp. 71–93, Mar. 2006. (Cited on page 8.)

M. M. Wanderley, “Gestural control of music,” in *International Workshop on Human Supervision and Control in Engineering and Music*, 2001. (Cited on page 8.)

D. Wessel, “Instruments that learn, refined controllers, and source model loudspeakers,” *Computer Music Journal*, vol. 15, no. 4, Dream Machines for Computer Music: In Honor of John R. Pierce’s 80th Birthday, pp. 82–86, 1991. (Cited on page 1.)

G. Widmer, S. Dixon, W. Goebl, E. Pampalk, and A. Tobudic, “Horowitz Factor,” *AI Magazine*, pp. 111–130, 2003. (Cited on page 14.)

J. O. Wobbrock, M. R. Morris, and A. D. Wilson, “User-defined Gestures for Surface Computing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’09. ACM, 2009, pp. 1083–1092. (Cited on page 146.)

Y.-j. Wu and R.-h. Wang, “Minimum Generation Error Training for HMM-Based Speech Synthesis,” *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, vol. 1, pp. I–89–I–92, 2006. (Cited on pages 24 and 111.)

J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, “Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 66–83, Jan. 2009. (Cited on page 25.)

E. Yamamoto, S. Nakamura, and K. Shikano, “Speech-to-lip movement synthesis based on the EM algorithm using audio-visual HMMs,” in *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, 1998, pp. 2–5. (Cited on page 27.)

H.-D. Yang, A.-Y. Park, and S.-W. Lee, “Gesture Spotting and Recognition for Human–Robot Interaction,” *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 256–270, Apr. 2007. (Cited on page 52.)

S. Z. Yu, “Hidden semi-Markov models,” *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010. (Cited on page 69.)

B. Zamborlin, “Studies on customisation-driven digital music instruments,” PhD Dissertation, Goldsmith University of London and Université Pierre et Marie Curie, 2015. (Cited on page 49.)

B. Zamborlin, F. Bevilacqua, M. Gillies, and M. D’Inverno, “Fluid Gesture Interaction Design: Applications of Continuous Recognition for the Design of Modern Gestural Interfaces,” *ACM Transactions on Interactive Intelligent Systems*, vol. 3, p. Article 22, 2014. (Cited on page 37.)

H. Zen, K. Tokuda, and T. Kitamura, “Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences,” *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, Jan. 2007. (Cited on page 24.)

H. Zen, Y. Nankaku, and K. Tokuda, “Continuous Stochastic Feature Mapping Based on Trajectory HMMs,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 417–430, 2011. (Cited on pages 3 and 26.)

H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2013, pp. 7962–7966. (Cited on page 25.)

L. Zhang and S. Renals, "Acoustic-Articulatory Modeling With the Trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008. (Cited on pages 3 and 26.)

A. Zils and F. Pachet, "Musical mosaicing," in *Digital Audio Effects (DAFx)*, vol. 2, 2001. (Cited on page 135.)

Colophon

This work was supported by the ANR project Legos (11 BS02 012).

Final Version as of June 9, 2015.
Jules Françoise: *Motion-Sound Mapping By Demonstration.*