CO/DA: Live-Coding Movement-Sound Interactions for Dance Improvisation

Jules Françoise
jules.francoise@lisn.upsaclay.fr
Université Paris-Saclay, CNRS, LISN
91400, Orsay, France

Sarah Fdili Alaoui
saralaoui@lri.fr
Université Paris-Saclay, CNRS, INRIA, LISN
91400, Orsay, France

Yves Candau
ycandau@sfu.ca
Simon Fraser University, School of Interactive Arts + Technology
Surrey, BC, Canada

Figure 1: Dancers improvising with the live-coding environment CO/DA.

ABSTRACT

We present a performance-led inquiry that involved a live coder programming movement-based interactive sound and two dance improvisers. During two years of collaboration, we developed a joint improvisation practice where the interactions between the dancers' movement and the sound feedback are programmed on the fly through live coding and movement sensing. To that end, we designed a new live coding environment called CO/DA that facilitates the real-time manipulation of continuous streams of the dancers' motion data for interactive sound synthesis. Through an autoethnographic inquiry, we describe our practice of sound and movement improvisation where live coding dynamically changes how the dancers' movements generate sound, which in turn influences the dancers' improvisation. We then discuss the value, potential and challenges of our dance/code improvisation practice, along with its implications as a design method.

CCS CONCEPTS

• **Human-centered computing** → **Interaction design**; **Human computer interaction (HCI)**; **Auditory feedback**; **Gestural input**.

KEYWORDS

Live Coding, Dance, Improvisation, Embodied Interaction, Movement, Design.

ACM Reference Format:
Jules Françoise, Sarah Fdili Alaoui, and Yves Candau. 2022. CO/DA: Live-Coding Movement-Sound Interactions for Dance Improvisation. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3491102.3501916

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501916

1 INTRODUCTION

With the third wave of HCI, interest in exploring more expressive and embodied forms of interaction has spawned the development of many methodologies and tools for the design of movement-based interactions, covering a wide array of applications such as health and wellbeing, gaming, or artistic practice. As a result, we have witnessed the emergence of design approaches based on bodily experiences, drawing from somatic and artistic movement practices [27, 42, 45, 49, 55, 77]. While several design approaches are explicitly inspired by dance, advances in interaction design and technologies have also significantly impacted the dance community for several decades [17, 64]. Indeed, a growing body of work in HCI results from experiments integrating technologies in art practice, and in dance in particular [26, 48, 68, 79]. These contributions have shown that there is much to be learnt from performance as an experimental ground to study how people interact with technologies. Benford et al. described these approaches as performance-led research in the wild [8]. We follow this approach and present a performance-led inquiry that involved two dance improvisers and a live coder who programs movement-based interactive sound. During two years of collaboration, we developed a joint improvisation practice where the interactions between the dancers' movement and the sound feedback are programmed on the fly through a custom live coding platform, as illustrated in Figure 1.

Technology-mediated improvisation practices have been developed in other fields than dance, in particular music and visual arts. New Interfaces for Musical Expression (NIME) can be designed with an explicit focus on improvisation, for instance using custom controllers based on gestural or full-body interaction. The artistic practice of *live coding*, which consists of interactively programming musical or visual processes as performance [11, 20, 52], gave rise to new programming languages [4, 57, 73] and unique reflections on improvisation [18, 54, 71, 72]. Yet, approaches to live-coding addressing movement analysis and sonification remain scarce.

To support our improvisation practice, we designed a new live coding environment called CO/DA that facilitates the real-time manipulation of continuous streams of the dancers' motion data for interactive sound synthesis. The movements of one or several dancers equipped with sensors are streamed to CO/DA, allowing the live coder to program the interactions between their movements and the sound feedback on the fly. Our approach involves capturing, modeling and sonifying a dancer's movements in real-time during the performance, leading to a joint improvisation. Our design is autobiographical in that it is built for ourselves based on own interest and for our own practice [58].

Through an autoethnographic inquiry, we describe our practice of sound and dance improvisation where live coding dynamically changes how the dancers' movements generate sound, which in turn influences the dancers' improvisation. We then discuss the value, potential and challenges of our dance/code improvisation practice, along with its implications as a design method.

Our contribution in this paper is twofold. First, we present our artistic practice from a first-person perspective, built upon the autobiographical design of a live coding environment for movement-sound interaction along with an improvisation method. Second, we describe what we learnt as designers and performers from our practice of live coding with sound and movement improvisation, using an autoethnographic methodology. Finally, we contribute to a broader audience in HCI with reflections on the potential and challenges of live coding as a design practice for embodied interaction.

2 RELATED WORK

Our project is rooted both in embodied interaction design approaches developed in the HCI community, and in artistic practices from dance and music communities. This section outlines key related work in designing movement-based interactions, technologies for dance and choreography, and live coding as an improvisation practice.

2.1 From Dance to Embodied Interaction

There is growing body of work in HCI investigating how movement practices can inform and support interaction design. Approaches such as Move to get Moved [45], Moving and Making strange [49], Embodied Sketching [55], or somaesthetics [42, 67] acknowledge the primacy of the bodily experience or "the soma" in all the steps of the interaction design process. The vision of designing "for movement and through movement" has shown its potential to enhance awareness to one's self and quality of attention [77] for both designers and users. Such contributions, both methodological and technological, tend to draw inspiration from somatic practices and dance in order to offer novel and valuable designs. Examples of research that drew from dance to enrich interaction by taking into account user's expressivity include studies focusing on the notion of movement qualities [27] or Laban Efforts [28].

Complementary to contributions to design methods, there is growing interest in creativity support tools applied to dance annotation [13], learning [64], documentation [17] or enhancement of kinesthetic creativity [44]. Thus, historically many of the HCI works linking dance and technology emerged from either explicitly drawing inspiration from dance to contribute with novel interactions and HCI concepts or using design principles to benefit the dance community through new interfaces and interaction capacities.

2.2 Dance, Performance and Improvisation in HCI

An extensive literature review of the research in dance and technology throughout the past two decades that covers publications from SIGCHI and from the movement and computing (MOCO) conference is now available [79]. It summarises important themes in the literature and trends in the HCI and dance research community. This review only cites few examples of research in HCI that venture in the world of dance productions or what goes under the umbrella described by Benford et al. as performance-led research in the wild [8]. Benford et al. argues that performance can act as an experimental frame to study how people interact with technologies in the wild, in real performance settings outside of the lab. Such projects triangulate artistic practice with studies to understand people's experience and articulate theory from the resulting concepts and frameworks. Among the contributions that emerged from studying dance performance in the wild, we can cite Neural Narratives by Bisig and Palacio where an interactive visualisation responds to the dancer's movement to encourage novel movements during improvisation [10]. In a performance by Eriksson et al., drones are incorporated into the classic opera of Medea, and acted as Medea's children. The authors discuss through this performance the intercorporeality between the human and the drones [25]. Lately, Fdili Alaoui and Matos created a piece called RCO where audience members participate to the piece following instructions on their mobile phone [5]. The authors studied how mobile technology contributed to both coerce and liberate the participants in the piece.

Such dance performance-led research in the wild remains scarce in HCI. This might be due to the fact that art-making is usually personal and driven by individual goals and intentions and thus generates insights that are hard to generalise or sometimes to articulate in academic forms and languages. Fdili Alaoui discussed how the integration of technology in her dance piece SKIN [26] created tensions but also knowledge on the messy nature of artistic intent and on the high requirements of stage-level robustness. She emphasizes the difficulty and the challenge for HCI to stay open to the singularity of art practice but encourages to welcome contributions gained from the knowledge potentially generated through art-making that stays faithful to the practitioner's individuality.

2.3 Movement-Based Interaction Design Software

With the growing availability of cost-effective sensing devices, technologies supporting the design of movement-based interactions have evolved over the years to integrate novel methods and user interfaces. The pioneering work of Camurri et al. with Eyesweb [14] has had a sustained influence in both scientific and artistic research communities. Eyesweb provides a visual programming language for building movement analysis pipelines that go from low-level motion descriptors to high-level analysis of movement's expressive or affective qualities [15, 59]. It also integrates machine learning algorithms for gesture recognition [35]. Other visual programming platforms such as Cycling'74 Max/MSP [3], PureData [62] or vvvv [2] are popular with creative communities, in particular in computer music and the performing arts. Some of them include specialized tools for movement interaction design. MuBu is Max package dedicated to the management of multimodal data, movement signal processing and movement sonification [63, 69]. It integrates the XMM library for user-centered machine learning [32], enabling musicians and artists to design movement-based interactions by demonstration. Such user-centered approaches to machine learning have become popular to rapidly build custom gesture recognition and sonification systems. Examples include Fiebrink's Wekinator [30], which is widely used in music performance and pedagogy, the Gesture Recognition Toolkit [36], and more recently the RapidMix API [9] and InteractML [23].

While these tools are dedicated to facilitating the design of movement-based interactions, they focus on rapid prototyping rather than live coding and improvisation. Although live editing is possible with Max, PureData or VVVV, we found in early experiments that visual programming could be tedious when building complex movement analysis pipelines in an improvised manner, compared to textual environments that support commenting, functional abstraction mechanisms and execution on demand.

2.4 Live Coding

The long and complex history of improvisation in music and the performing arts has stemmed the development of a variety of practices with technology. Many approaches to improvised electronic music have emerged withing the New Interfaces for Musical Expression (NIME) community, for instance with movement-based on gestural instruments [47, 70], algorithmic processes and artificial agents [6, 60, 61] or laptop orchestras [74, 76]. A particular thread of artistic practice and research has been looking at improvisation through the manipulation of computer code, leading to the "live coding" movement in music and video [11, 20]. Frustrated by the rigid interfaces offered by music production software, they started engaging with programming musical processes on the fly, using custom software and programming languages. Computer programming as performance has flourished, giving rise to a large variety of practices [19], software and programming languages [4, 53, 57, 75]. Live coding has been discussed as an improvisation practice that engages with the audience, in particular through the projection of the code during performances [54], as stated in the TOPLAP manifesto: "Obscurantism is dangerous. Show us your screens.".[1] Through the affirmation that "algorithms are thoughts", the manifesto highlights a fundamental desire of practitioners to engage with algorithms, not only with code, during improvisations [18, 54].

Live coding is not limited to programming music and audio processing, and several practices involving dance have emerged. Nick Collins' experiments with Teresa Prima involved changing instructions for the performer using a chalk board [18]. The most prominent figure in algorithmic choreography and live coding is Kate Sicchio. In her practice, the instructions contained in the code displayed to the performers and audience are interpreted by humans rather than computers. With "Hacking Choreography", Sicchio investigated how manipulations of the dance score could "circumvent, subvert, or repurpose the original choreography" [71], exposing the subtleties and cognitive challenges for dancers to perform live choreography. Beyond textual instructions, she designed the live coding tool Terpsicode to generate choreographic patterns using sequences of images [73]. Her collaboration with Alex McLean "Sound Choreography <> Body Code" [72], involved a joint improvisation between choreography and music where the dancer's movement, captured with a Kinect, interfered with the music notation, while the dance score was itself modified by the live coder's sound. Their collaboration was about confronting two practices involving notation, in dance and music, with processes influencing each other. However, their work focused on choreography rather than dance improvisation, and it did not involve live coding the movement sonification itself.

2.5 First-Person Methods in HCI

First-person methods have a long history in the humanities and social sciences, and were introduced lately in HCI and Interaction Design. First-person methods in HCI include autoethnography, autobiographical design and research-through-design [21, 51].

Autoethnography is a research method that uses personal experience to describe and interpret cultural experiences, beliefs, and practices [24]. In HCI and design, it enables researchers to articulate experiences of a design, a prototype or a concept from within, using themselves as the subject of the study. For example, Lucero presented an autoethnography of their own experience of living without a mobile phone over a period of nine years, showing the social, professional and personal impact of mobile technology removal [50]. Later, Homewood et al. applied an autoethnographic method to show how the removal of technology in their menstrual cycle tracking facilitates emotional, embodied and cultural knowledge of their lived experience of self-tracking [40]. Jain et al. presented an autoethnographic study of their travel as a hard hearing individual [46] which led to design explorations of personalized technology for aiding their travel and more broadly for deaf and hard of hearing users. The value that these autoethnographic methods have, beyond generalizability, is the extent to which their insights resonate critically with the readers' personal experiences and understanding of interaction [24].

[1] The Temporal Organisation for the Parsimony of Live Art Programming (TOPLAP) was founded in 2004 to explore and promote live coding. Its manifesto, along with other ressources on live coding is available online: https://toplap.org/

Autobiographical design and research through design focus on using personal experiences within the design process. With "the video window", Gaver described their own design, aesthetic choices and personal experiences of living with a video screen hanging in their bedroom wall, that displays the image of the skyline from outside [33]. Later on, Neustaedter and Sengers introduced autobiographical design more formally and argued for the value of developing a systematic way of designing with and for the self [58]. Desjardins and Wakkary described their project of converting a Mercedes Sprinter into a camper van [22]. Through their autobiographical design, they offered a rich reflection on the making of their personal space. Heshmat et al. used an autobiographical method to design 'Moments', an always-on video recording system used by one of the researchers and their family over a two-year period. Their design shed light on the family's experience of being captured, commitment to keeping the system running, and privacy issues [39].

First-person methods in HCI have raised questions regarding their validity and generalizability. According to Zhang and Wakkary, it is undeniable that designers apply and benefit from their personal experiences in their design practice [78]. They argue for recognizing the legitimacy of designers' experiences in interaction design, whether it concerns their observation of real-life events or their interaction with design artifacts and systems [78].

Experiments and designs that take the stance of first-person methods show the relevance of researchers' and designers' personal experiences for designing and gaining in-depth and long term knowledge on the interaction with systems within the field. Our work aligns with such methods, their values and their benefits. We present a personal artistic practice of live coding and dance improvisation where we designed a live coding technology for ourselves in an autobiographical manner and gained insights on our embodied experiences with it from an autoethnographic study covering 2 years of extensive studio time together.

3 THE ROOTS OF THE PROJECT

3.1 Team Members

This practice was born of the desire of the three co-authors of the paper to create an artistic and a research project that combines our sensibilities and interests. All of us have extensive past experience working with interactive sound and movement sensing on stage. Two of us are professional dance artists that perform regularly on stage and wanted through this project to work on dance improvisation with sound. One of us has an extensive background in music and technology and particularly in interactive machine learning and movement sonification. All three of us are HCI researchers investigating embodied, movement-based and sound-based interactions. The research and the creative process was collaborative from the beginning, and all artistic decisions, research and technical questions emerged organically from the improvisation sessions. The paper carefully narrates our own artistic and intellectual journey with our design and artistic practices.

3.2 Initial Intentions

Our live coding and dance improvisation is rooted in a previous project called *Still, Moving* where we researched how to design auditory feedback for micro-movements in the context of an interactive installation and dance performance [31]. Our design process drew upon regular studio sessions to foster improvisation and exploration, which allowed for the emergence of unexpected mapping strategies. The complexity of such iterative process led us to reflect on how current tools and methods did not allow for a design of the sound feedback that adapts to an improvised – and thus an evolving – movement material.

Subsequent iterations of our previous project resulted in an improvised dance performance that we performed in 2017 in London at MOCO'17. In this performance, we experimented with real-time gesture recognition: the gestures to be recognized were not predefined, but captured live during the performance. This scenario was based on an interface that we specifically implemented for the performance but which flexibility remained limited.

This led us to envision a practice of live coding where movement and technology-mediated interactions would be jointly improvised through the collaboration of dancers and programmers. Our typical interaction scenario is illustrated in Figure 2. It involves one or several dancers, whose movements are measured using motion sensors, and one live coder, who processes motion signals to generate feedback in real-time. Both the movers and the live coder start with a blank slate, and they develop an improvisation where sound is generated from the dancers' movements, according to interaction patterns programmed on the fly by the live coder. This creates a multitude of feedback loops: the sound feedback influences the movement improvisation, which in turn inspires the coder to further alter the relationships between movement and sound. This network of interactions and negotiations leads to highly dynamic improvisation sessions that stimulate the exploration of a large number of movements and designs scenarios.

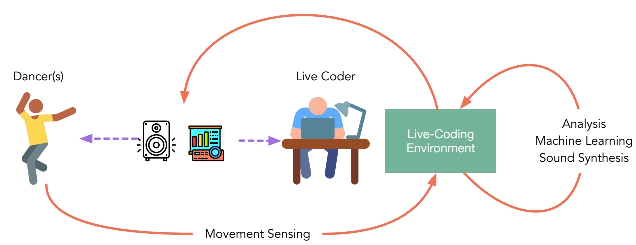

Figure 2: Our practice involves a joint improvisation between one or two dancers and a programmer.

3.3 Improvisation Sessions

Our approach involves a constant interplay between practice, theory and evaluation. For over two years, we engaged in improvisation sessions aimed to facilitate the experimentation of new embodied ideas and strategies for movement-sound interaction. These sessions involved one live-coder improvising with either one or two dancers simultaneously. Improvisation sessions served as material for a theoretical reflection on both technological constraints (the characteristics of the live coding environment) and aesthetic constraints (the mapping strategies employed). By integrating the outcomes of each session rapidly into the live coding environment, theory fed back into practice.

This improvisation practice radically differs from typical approaches to artistic rehearsals with technology. Usually, artefacts are developed iteratively by alternating phases of ideation, prototyping and evaluation. This approach can limit the possibilities of expression: prototypes are often developed and tested using the developer's gestures, which do not match the expressive abilities of an expert dancer, therefore limiting the relevance of the system. Furthermore, *evaluating* a prototype often means judging the system rather than exploring its potential to generate unexpected outcomes. In order to allow the *emergence* of new interactions between movement and sound, we progressively developed a set of structures, rules and exercises to foster exploration on the fly. Some key principles of our improvisation sessions are outlined thereafter:

(1) Verbal communication between the dancers and with the live coder is not allowed during improvisations, in order to encourage embodied risk-taking and to allow the resolution of problems encountered through the improvisation itself, moving through it or coding it – rather than by interrupting the improvisation to verbalize limitations.
(2) Failure is not a reason for interruption. Several types of failure can (and do) happen over the course of most improvisations: bugs in the software, audio glitches due to wrong parameters, poor sound design or meaningless motion-sound coupling. Solutions should be found in the moment through movement or edits to the code.
(3) Improvisation exercises can stimulate the discovery of new relationships between movement and sound. The exercises can, for example, encourage the creation of gestures or postures (by incrementally defining a vocabulary during improvisation), stimulate the generation of counter-intuitive movements (by relying on non-trivial relationships between different body parts), or focus on the body's response to particular sounds.

With such principles, improvisations involve a complex set of interactions and negotiations between the dancer and the programmer that are fruitful to stimulate new movement patterns and design ideas.

4 DESIGNING CO/DA: A LIVE CODING ENVIRONMENT FOR MOVEMENT-SOUND INTERACTIONS

We developed a software library and a live coding environment called CO/DA, dedicated to live coding of the interactions between motion and sound. The approach differs from existing live coding environments for algorithmic music generation, which often focus on the generation of pulsated music sequences. Our approach to movement sonification emphasizes continuous interaction: streams of motion data from wearable sensors are analysed and processed to control sound synthesis continuously and in real-time. We focus on sample based synthesis technique (granular synthesis and concatenative synthesis) where synthesizers are controlled by continuous parameters to create rich sound textures, rather than by sequences of notes to produce music.

We start by presenting design principles and our implementation of the live coding environment in its final version. Although its capabilities have evolved over practice, its requirements and architecture were identified early in the project. We then present some of the iterations of our practice-driven development process that contributed to shape the features of the software.

4.1 Design Principles

In our earliest experiments with live coding, we prototyped a software library for Max/MSP where pipelines could be created by composing operators using visual programming. However, visual programming requires creating boxes and drawing connections with the pointer to create pipelines, which we found tedious and time-consuming when complex movement analysis pipelines are required. Moreover, duplicating and modifying pipelines is less convenient with visual programming than with textual programming languages supporting a functional paradigm. Finally, edits to the patcher in Max resulted in audio interruptions due to changes in the audio processing graph. From these experiments and the observed limitations, we derived a set of requirements for a live coding environment.

First, we opted for **textual programming**, with **interactive execution**. Instructions and code blocks should be executed on demand by the programmer through keyboard shortcuts, as found in other live coding environments such as Supercollider [56] or TidalCyles [57]. Second, we decided to base CO/DA on an **existing programming language**. While there exist a number of domain-specific languages dedicated to live coding, other environments rely on scripting languages like JavaScript [65]. Third, CO/DA should be **designed for continuous movement-based interaction**. This demands that the environment provides built-in utilities for sensor data acquisition, event stream management and signal processing. Finally, the environment should provide **visual feedback** on various streams of motion data for monitoring.

Our reflection on these initial specifications led us to establish the following design principles for the architecture of CO/DA.

Event-Driven Architecture. Designing interactions based on continuous movement data involves manipulating asynchronous event streams that originate in various sensors or motion capture systems. Support for data stream processing should be provided across the live coding environment, from movement data acquisition and analysis to sound synthesis.

Functional Motion Signal Processing. Crafting movement-sound interaction often requires the creation and fine-tuning of movement processing pipelines that extract features from raw data. We propose that a functional approach to signal processing provides flexibility for (re-)defining such pipelines on the fly.

Interactive sound synthesis. Our interest in movement sonification requires that the live coding environment integrates sound synthesis engines that can be continuously driven by streams of parameters. Sound synthesis engines should be carefully selected and provide a control interface compatible with data streams.

Based on the Web. Web technologies facilitate distribution, sharing and collaboration. Basing a live coding environment on the web allows for easily distributing environments and applications to a larger audience.

4.2 Implementation

CO/DA is composed of a set of libraries and a live coding environment implementing the above principles. CO/DA is distributed as open-source software and is available online.[2] In order to facilitate adoption and to enable its deployment on multiple platforms (including mobile), it has been programmed as a JavaScript library and web application using standard web technologies.

Core Libraries. Asynchronous event streams are the core construct in CO/DA. They are used to represent time-varying processes such as sensor data, motion features or sound control parameters. Our implementation is built upon a reactive programming paradigm [7] enabling rapid and flexible processing of continuous movement data. It facilitates the creation, filtering, transformation and consumption of asynchronous data streams that propagate changes over the interaction pipeline. Our implementation of reactive streams relies on Most.js, a high performance reactive programming library [1].

CO/DA has a modular architecture based on streams, components and operators. Components are JavaScript objects that expose a set of reactive streams as properties. Operators are functions applied to streams for filtering or transforming their events. This architecture makes the library extensible with new operators, input devices or output engines.

The core libraries provide:

- Input components to acquire data from different sensors, in particular Myo Armband, Leap Motion, R-IoT or smartphones (through a dedicated web page).
- Motion analysis modules implemented as functional operators acting on data streams. These include basic mathematical operations, filtering, scaling, derivation, or more advanced signal processing techniques such as wavelet analysis.
- Real-time data visualizations of the motion parameters. These visual interfaces are created on the fly and can apply to any numerical data stream.
- Machine learning modules for dimension reduction (PCA), gesture recognition and mapping (GMMs, HMMs). The machine learning modules are fully reactive, it is therefore possible to define a movement vocabulary incrementally during an improvisation.
- Sound synthesis engines and audio effects which parameters can be continuously driven by reactive streams. These components include granular synthesis, concatenative sound synthesis, and digital audio effects, based on the Web Audio API.

Live Coding Environment. The core libraries can be used to build standalone web applications for movement sonification. Additionally, CO/DA provides a live coding environment that runs in a web browser and allows the programmer to dynamically execute portions of code and visualize data streams. Our implementation integrates a sandboxed environment that automatically tracks the execution of reactive streams. Keyboard shortcuts enable the coder to execute instructions or entire code blocks. The set of instructions is then parsed to execute or replace reactive streams on the fly. A screenshot of the live coding environment is displayed in Figure 3. A hosted version of the live coding environment is also available online [3].

4.3 Practice-Driven Developments

We now present three selected iterations in the development of the project. We start by reporting results from practice sessions, then we report on their implementation and provide reflections from practice.

4.3.1 From Movement to Sound. We first started by establishing an API facilitating the creation of pipelines linking movement data to the sound synthesis. This API was iteratively refined during the first improvisation sessions, from the live coder's experience. We chose to use a fluent API facilitating the composition of operators over reactive streams. CO/DA's API is illustrated through the code example available in Figure 4. In most of our practice, we used Myo armbands for movement sensing for it is easy to use and provide both inertial measurements and electromyography (EMG). In this example, the EMG data from a Myo armband is analyzed to extract an estimate of the force exerted through muscular contraction. The overall force (summed over all channels) is then continuously mapped to the position in the audio file to drive the granular synthesis.

```
// Connect to the default myo armband
m = myo();

// Plot the acceleration data stream from the IMU
acc = m.acc.plot();

// Plot the EMG data and compute an estimate of the force
emg = m.emg.plot().force().mvavrg({ size: 9 }).plot();

// Create a granular sound synthesizer
synth = granular({ file: 'hendrix' });
synth.connect();

// Map the overall force to the position in the audio file
synth.position = emg.sum().autoscale().plot();
```

Figure 4: Code snippet illustrating simple movement analysis in CO/DA.

We started with direct mapping strategies, evolved toward many-to-many mappings before experimenting with more indirect forms of interaction. Moving away from overly transparent strategies to stimulate movement beyond habits, we experimented with inversion mechanisms and counter-intuitive mappings. For instance, instead of associating the movement's energy with the loudness or density of sound, we inverted their relationship so that a deceleration of the movement would lead to accelerating sound. This generated a stimulation for phrasing movement in sequences, each gesture calling for a response and continuation. Applying the same process to the distance between limbs, or to their relative velocity, created further exciting interactions leading dancers to experiment with symmetry and asymmetry, synchronization and dispersion.

[2] Online documentation: https://codajs.netlify.app/
Source code: https://github.com/JulesFrancoise/coda

[3] Hosted version of the live coding environment: https://playcoda.netlify.app/. Note that most of the input devices require running the environment locally.

Figure 3: Screenshot of the CO/DA live coding environment. The left panel is a text editor enabling interactive execution. The right panel is used to display real-time visualizations.

Layering or combining mappings led us away from control towards indirection between movement and sound, fostering more engaging improvisations.

4.3.2 Scaling Signals. During the first sessions, the issue of properly scaling motion signals made it difficult for the coder to develop sonification strategies on the fly. While obvious, the need for scaling differs in live coding practice compared to more traditional prototyping approaches where the vocabulary and range of movements is established in advance and more time can be dedicated to fine-tuning. During improvisations, not only is it difficult to evaluate a range of variation from observation, the movement vocabulary can dramatically change over short periods of time. A dancer jumping might inspire the coder to create a particular mapping, only to find the dancer moving slowly on the floor when the code is ready for execution.

These challenges led us to design several strategies for facilitating the scaling of movement signals. We developed an `autoscale` operator that keeps track of the observed min/max values to provide a normalized version of the input stream, no matter the amplitude of the dancer's movement. A variation of this strategy was found with an adaptive mapping where the stream is scaled according to the min/max values on a sliding window [31]. With this strategy, the range of variation progressively adapts to the dancer's range of motion over a given time period.

These operators were heavily used, especially in the early stages of our practice. With more experience, the live coder started integrating a more tacit knowledge of the range of variations of particular sensors, and started hard-coding scaling parameters to known values for particular movement patterns. Yet, he continued to use the adaptive scaling operator that embeds a non-linear behavior found stimulating for improvisation.

4.3.3 Mapping Intention and Observation with Machine Learning. One of the challenges of joint dance/code improvisation is the heterogeneity of time scales accessible to the performers. While dancers can instantly shift the way they move, programming new interactions cannot be planned and executed in seconds. This asymmetry creates a tension between the wish to observe the dancer and draw inspiration from the improvisation, and the actual capacity to capture their movements and intentions. In particular, several of the coder's intentions were hard to program by hand. First, extracting the meaningful features of particular movement patterns remains challenging, especially when the characteristics relate to movement qualities. Second, capturing specific gestures that emerge and recur through the improvisation can hardly be coded manually.

To bridge this gap, we studied how interactive machine learning methods could enable the code to record, analyze and recognize movement patterns on the fly. We implemented several machine learning algorithms that have previously been used in the computer music literature [30, 32]. These techniques share a common architecture based on a simple recording interface allowing the coder to capture and annotate movement excerpts. Using reactive programming, any change to the recordings is propagated to the learning algorithm, that trains a new model from the updated data in a background web worker. The prediction engine is then seamlessly updated with resulting model, without interruption in the processing. A code example with the associated visualizations is presented in Figure 5.

We implemented two types of models following different learning paradigms:

- **Unsupervised learning** , in particular Principal Component Analysis (PCA), was used to extract meaningful dimensions from a set of motion descriptors. This dimensionality reduction technique identifies linear combinations of dimensions that maximize the variance, thus generating descriptors for particular movement patterns.
- **Supervised Learning** enables real-time gesture recognition from a set of labeled recordings, using Gaussian Mixture Models (GMMs) or Hidden Markov Models (HMMs). During prediction, the model outputs streams of each gesture's likelihood, which can be used to drive the sound synthesis.

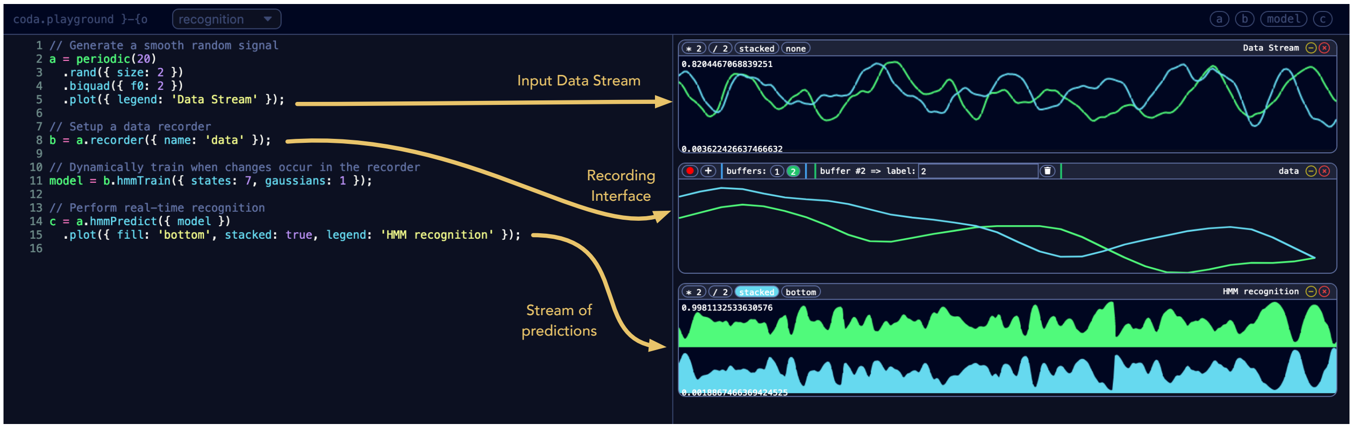

Figure 5: Screenshot of the live coding environment with an example of gesture recognition with Hidden Markov Models. The left panel shows the text editor with the code necessary to (1) create a smooth random signal emulating motion data, (2) create a recorder, (3) create a model that trains when changes occur in the data, and (4) compute predictions from the input stream with the model. The user interfaces associated with data visualization and recording are displayed in the right panel.

Our experiments with feature extraction was stimulating for quickly generating arbitrary mappings. However, as the live coder's expertise grew, crafting analysis pipelines and custom relationships became easier and analysis based on PCA was abandoned. Supervised learning approaches for gesture recognition or mapping were found particularly interesting for they can capture specific gestures executed by the dancer during an improvisation. This way, a movement vocabulary is constructed on the fly through the convergence of the dancer's expression and the coder's observation. Yet, these methods remain challenging to use in performance because they require that training examples are well segmented, meaning that the coder needs to anticipate the dancer's movement to record meaningful gestures.

5 LESSONS LEARNT FROM OUR PRACTICE OF LIVE CODING WITH SOUND AND MOVEMENT IMPROVISATION

In this section we describe our experimental methodology and we report on findings that emerged from a large number of improvisation sessions.

5.1 Experimenting in the Studio : Methodology

5.1.1 Procedure and Participants. We followed a first-person autoethnographic method making use of a story-based approach to gather data on our own personal embodied experiences of the improvisation with live coding [41]. Thus, the participants to our experiments are ourselves, the three performers and authors of the paper.

The improvisation sessions took place in the studios where we rehearsed. At the end of each improvisation we gathered in the center of the studio to discuss our experience of the session. Through these accounts of our own experiences within an autobiographical design process, we do not claim to produce generalisable findings. Instead, we aim to provide an understanding of how we designed a live coding technology that was the basis of our long-term artistic practice. Our autoethnographical accounts describe personal embodied experiences captured throughout two years of practice and design. Their value is to inform the community on an intimate take on the technology experienced from within while inspiring novel design directions beyond our specific use.

5.1.2 Data Collection and Analysis. We analyzed about 23 improvisation sessions performed over 13 days, between June 2018 and April 2019. Each improvisation and discussion was recorded using a camera placed at the back of the studio. The improvisations lasted from 10 minutes to 1 hour. The discussion lasted from 10 to 30 minutes. Each of us took personal notes during the discussions. The discussions were facilitated informally: they started with us checking on each other and then followed the flow of the conversation.

Two of us (the live coder and one of the dancers) observed the video recordings of the sessions and transcribed the significant events that happened in each one. We also transcribed our notes. We then analyzed our transcriptions using a thematic analysis approach [38]. We defined concepts using our original words (open coding) and grouped them in categories (axial coding). We verified and discussed our analysis with the third author (one of the dancers) who previously read the transcriptions to ensure that the analysis captured the data.

5.2 Finding the Sweet Spot between Ambiguity and Control

The degree of clarity in the relationship between the performers and thus between the movement and the produced sound varied in a scale that went from total ambiguity to total control. Exceeding ambiguity arose when the sonic space was too layered and complex or when the movement was too fast or changed too much. That led the performers to perceive the mapping as non-readable. On the other hand, performers experienced improvisations as overly controlled when the mapping was too simple, direct and predictable.

When starting from a blank slate, sessions often began with obvious mappings, inviting the dancers to start with simple movements and to explore repetition and sobriety in the body. However, such simplicity also produced limited improvisation capacity. After exploring repetitive movements that consistently triggered the same sound, one of the dancers reported that "the body is not a keyboard". She was looking for a progression in the complexity of the mapping and for variations in the range of possible modulations of sounds that she could produce with her body. On the other hand, ambiguous mappings often occurred at the end of the sessions, when the live coder had layered algorithmic processes linking the sound and the movement, creating incomprehension for the dancers.

All three performers expressed their experience of being in the flow of the collective exploration and in connection with each other in a memorable session that lasted 1 hour of uninterrupted improvisation, without speaking nor stopping. They reflected on the quality of the mapping during that session by calling it "a sweet spot" when "it's readable but surprising and ambiguous at the same time. that creates material that I can play with for a long time". According to the live coder, "the performance becomes much more organic [...] when I look at you two, you seem together, connected. You are not looking at the screen – which is a good sign, – [...] you are in the flow of the improvisation". On the organicity of dancing within a sweet spot between control and ambiguity, one of the dancers noted: "when we are in that sweet spot, I hear what I produce, what the other dancer produces, I become more perceptual, more sensitive to myself and to my partners. It takes much less energy to do it, the body is already engaged in it without too much effort. It's more embodied." However, the right dose of control versus ambiguity seems to be the golden ratio that is difficult to find and precarious to maintain. One of the dancer expressed her feeling of being disembodied and disconnected when that ratio is lost : "when I don't have that right dose, I go back to my head, I am looking for something, I completely decapitate myself and I loose the connection to my body. It takes me a lot of energy to ask my body to dance."

The specificity of our improvisation is to play and practice with the complexity of a feedback loop where movement drives the sound, which feeds back into movement. One of the dancers expressed the indirection of this coupling: "Understanding the link between the body and the sound is not so important after all. What matters is to find a connection between us. We are not trying to understand how to produce specific effects of the technology, although technically that's what we are doing, for me this artistic practice is not cognitive, it's about feeling the connection to the two other performers. It's about enjoying possibilities for the body and for the sound."

5.3 Navigating through Persistence and Rupture

In most of our sessions, we observed that the performers played with repetition and sustained exploration of particular patterns. These phases of persistence were modulated by moments of rupture that would allow to break the flow of the improvisation, introduce surprise and thus avoid to bore the body into an endless redoing of the same movement.

According to one of the dancers, "there is value in trying something out, nothing happens, continue trying, repeat until you work something out. So in this practice I became very fond of repetition rather than innovation". Therefore, repetition of movement with the persistence of the same sound was enjoyable as it allowed to deepen the dancer's exploration of the movement-sound space. It helped the dancers modulate their movement and playfully find nuances within the space of possibilities. However, live coding being a time-based practice, the sonification evolves over the improvisation, creating moments of rupture that would push the dancers to explore a renewed space of movement sound relationships. One of the dancers said : "At the beginning of our explorations, the sound was produced as clear semantic events that I could perceive. If I repeated a gesture it reproduced the same sound. At this point I felt like I found the link. I did another gesture and it produced something else. That persistence was interesting to analyse what was going on. Then I could play with it. But eventually the live coder added a line of code and then the loop was broken. And this way, back and forth, we played between repetition and rupture". For the live coder, introducing rupture was interesting from an aesthetic point of view. It allowed to create surprises and to avoid the puppet effect or what he called "the mickey mousing" where the sound triggers are predictable.

Navigating through persistence and rupture means that performers experience a continuous learning and adaptation process involving discovery, exploration and appropriation. We analyzed the video recording of an improvisation session with one of the dancers, where a particular sonification based on energy accumulation led to non-linear audio response. The dancer made a gesture that is accompanied with a new sound. The dancer laughed out of surprise, then tried to perform the same gesture again, however without producing any sound feedback. He pursued the improvisation, exploring movement variations to progressively integrate an embodied understanding of the mapping enabling him to truly play. In response, the live coder continued altering this mapping strategy to modulate it over time and maintain the flow of improvisation. This example illustrates the processes at stake with such a dynamic practice. After an initial response of surprise, performers start experimenting and exploring movement variations to progressively and tacitly integrate the new action-perception loop until it is fully appropriated. By maintaining such a quality of attention, performers remain open to new forms of interactions, negotiating ruptures and unexpectedness through the improvisation rather than merely judging or evaluating the interaction.

5.4 Embracing Complexity and Ephemerality

In most of our improvisations, we reflected on the richness that comes from the way the dancers and the live coder influence each other. In our practice, we always start with blank code and silent bodies. Movement barely starts, then the coders starts populating the CO/DA platform. Little by little we build on layers of relationships. As time passes it becomes difficult to decipher them. The coder adds lines of code and with each line a different mapping is proposed or a variation is added. Each layer of code adds complexity to the mapping between the movement and the sound. Accordingly, the movement evolves, it generates the sound (through the sensing mechanism) and at the same time it responds to it.

Commenting on the session that lasted 1 hour of uninterrupted improvisation, the live coder expressed that "The richness of sound is a great indicator of how great was the movement improvisation. We feel that the sound doesn't really run out. As if all the performers are in some kind of question response as well as silences". The session was particularly memorable because it progressed seamlessly in adding layers of sonification. The first dancer said: "I start from nothing, I want to be autonomous, I am listening as much as I am doing, and then I see lines, and it becomes spaces, environments and then soundscapes, and it is enough to turn your head, I go here and I play with this, and Oh there is this sound too. It is possible to do very little and to hear a lot. That's what makes me explore silence and alternate with movement that have a lot of energy if I want to saturate the sound. It's like constant reward because no matter what you do there is something. It makes me always want to move."

There is an inherent complexity to an interaction scenario involving collective improvisation where none of the performer feels in total control of the interaction. From the dancer's perspective, the sound resulting from their movements is not entirely predictable, as it is ever changed by the programmer. From the live coder's perspective, tremendous uncertainty about the outcomes of his actions arises from the fact that programs are interpreted by another body. In fact, even when sessions started from the same mapping strategies, they always ended up in very different places.

With complexity comes ephemerality. Indeed, the combination of the live coder's choices with their interpretation by dancers create moments that are not reproducible. Each mapping and interaction scenario is only experienced once and then it vanishes with our movement. The poetics of our practice lies in this experience of being in the present. It is about creating a present and accepting to loose it. We embrace such ephemerality. We accept to loose what we create.

6 DISCUSSION

6.1 Designing through Feedback Loops

Our practice of improvisation with live coding can be thought as a design process continuously shaped by a multitude of feedback loops. Dancers' movements both act upon and are shaped by the sound feedback, which is itself created dynamically according to the inspiration that the live coder draws from observing the dance. Thus, moving with interactive sound feedback immerses the dancer in an action-perception loop that alters both their listening and their kinesthetic experiences [16, 37]. As such, improvised sonification offers reflexivity that goes beyond a paradigm of pure sound control: it guides and stimulates movement, inner bodily experiences and connections to others. The programmer is also immersed in a feedback loop that dramatically differs from standard prototyping approaches. While programmers often test their implementations by themselves during development, live coding implies that the created interactions are immediately explored and interpreted by another body. This network of interweaving processes creates a complex entanglement between the performers and favors the emergence of improvisational "Aha! moments".

Most design methods for embodied interaction tend to postpone engagements with technology. Approaches involving bodystorming and somatic practices often draw upon movement experience to stimulate design ideas, but implementation is deferred to a later time. Moreover, implementation and evaluation are often separated in an iterative design process. Because experiences are shaped by action-perception loops, we argue that designing *in action* within a network of entangled feedback loops offers new avenues for design methods where moments of surprise can emerge serendipitously.

Our approach was surely challenging. Indeed, relying on action-perception loops to design interactions on the fly with openness to uncertainty and serendipity can be risky. What if nothing happens? And certainly, there have been many improvisation sessions where the magic did not happen. We argue that there is no other way for such serendipitous interactions to occur than by taking the (long) time to be in the embodied and listening state that is necessary for these possibilities to form. Another challenge is that while the interactions happened on the fly, some technological choices needed to be made beforehand, which necessarily limit the range of the possible outcomes. However, within these constraints, our practice provides a unique opportunity for exploring the possibilities offered by a given apparatus beyond what is initially expected. Lastly, designing without talking can limit the amount of feedback that we communicate to each other on our experiences of the sonification and the dance. However, such a radical choice allowed us to avoid evaluating the designs through judgements which would break the flow of experience. We were cautious to be in an embodied state of listening to each other and to the sound, a state that can be precarious to maintain if verbalization is allowed. Instead, solutions are found by navigating through movement and code in a constant embodied negotiation that leads to a deeper exploration of the potential of our set up and limits particular designs.

6.2 Improvising Design

From our improvisation practice with live coding, we draw multiple design values that benefit the interaction design community beyond our specific use of CO/DA.

First, the results of our autoethnographic study revealed how we fluidly shift from persistence – through repetition of movement and consistency of the mapping, – to rupture. It also showed how we shift from moments of control of the movement-sound relationship to moments of ambiguity. Such shifts are only possible when we stay open to incidents and surprises as part of the creative process [34]. One of the values of our improvisation is thus the emergence of unexpectedness. Such emergence allows for critical experiences to happen and for the discovery of novel and unimaginable relationships. In addition to the openness to the possibility of unexpectedness, we embrace and play with complexity. Such complexity came from integrating a live coding technology in a series of improvisations with no other frame than spending a long time together, familiarizing our bodies with each other and with the tool, and observing how it frames our artistic endeavour. The shifts and the layers of complexity are part of the messy practice that we present. This echoes what has been shown by Hsueh et al. in their study of real world dance and music productions where they observed the fluidity and *"slippages"* that occur in the creative work [43]. They argued that existing creativity support tools tend to reinforce linear tasks and rigid roles and should instead account for shifts and slippages as they open up creative opportunities. Fdili Alaoui also argues for welcoming the messiness of how artists experiment with technologies in order to contribute to HCI with their own methods, insights and voices [26]. In line with these works, we propose to embrace messiness, complexity, and shifts (repetition and rupture or control and ambiguity) as conditions to mingle with artistic experimentation with technologies.

Another value of our creative practice that our autoethnographic study revealed is the openness to the ephemerality of the embodied and sonic experiences that emerged from our improvisation. Neither the scenarios that are created nor the interactions that they inspired were persistent nor were they stored for later use. And trying to reproduce them would be a utopia since the movement that would generate them is just as ephemeral. This value goes against how current technological innovation saves and stores our most mundane experiences. Designing with openness to ephemerality allows us to critically question the notion of persistence of experiences. Do we need to store and quantify all human experiences? who and what economical agenda does such practice benefit?

Our improvisation sessions took place during 2 years of regular practice with the live coding tool. Such a long term involvement of the group with the tool implied building a variety of skills starting from the capacity to attune to and deepen our kinesthetic empathy towards one another [31] and to fine tune our listening skills with regards to the possibilities of the sonification. Thus our improvisation practice with live coding is not about programming faster, innovating or generating design scenarios quickly. The process to learn and appropriate the technology and to build the sensitivities that are required to improvise together was a long journey. It is through a slow and long breath that we deeply explored the possibilities of sound and movement and enriched our creative experiences and design outcomes.

The opportunity to observe these explorations and skill acquisitions occurred because of our long term exposure to the tool. Thus, in line with Rogers and Marshall as well as Felice et al., we argue that studying technological artifacts in practice in the wild and on the long run allows to truly look at how users integrate them in their lives [66] and in their creative practice [29]. Certainly, as emphasized by Brown et al. [12] there are more risks, difficulty and messiness that comes from studying practices in the real world. This is particularly true with an autobiographical design and an autoethnographic methodology to assess it. We see richness in sharing our personal reflections, both as artists and as designers, that are sparked by our long-term experimentation with our tool. Our methodology, results and reflections genuinely reflects the time that we spent in the studio among each other and with the technology that we built for ourselves. A technology that at the end not only supported our artistic practices but defined it at its core.

6.3 Learning from a Personal Improvisation Practice

CO/DA is a personal project, where technology was designed together with a particular improvisation practice, for and through our personal experience. We believe that a broader HCI audience can learn from such subjective experiences. Our paper offers an authentic reflexive understanding of our design process and artistic endeavours which we hope will resonate with readers' experiences and critical thinking on technologies in art. Therefore, rather than generalizable findings or opportunities to replicate our work, we aim to provide concrete takeaways that interaction designers working in art, improvisation, sonification and embodied interaction can learn from.

First, designers should be open to the emergence of unexpectedness, complexity, ephemerality and shifts (between repetition and rupture or control and ambiguity) as conditions to experiment with technologies in art. Our improvisation practice offers opportunities to engage with designs in a way that promotes constant discovery, adaptation and learning. Second, we believe that designers can benefit from the exploration of entangled feedback loops, engaging early with technology and experimenting through shared in-situ experiences. Live coding brings the process of designing embodied interactions to a performance practice, which considerably affects the designer's experience in favor of risk-taking and on-the-fly problem solving. Such serendipitous approaches allow for suprises and emergence. Third, exploring non-verbal communication as a substrate for embodied design enables participants to enter a deep listening state and attunement to themselves and others. This constraint fosters rich experiences with early prototypes, and facilitates problem solving through interaction rather than judgment and evaluation. Finally, favoring long-term experimentation with interactive systems in the wild can help articulate how they can be used creatively in artistic practice.

7 CONCLUSION

We presented a performance-led inquiry tracing the emergence of an improvisation practice involving dance, movement sensing and sonification through live coding. Through an autobiographical practice-driven design process, we jointly created a live coding environment called CO/DA and a structured improvisation practice. Live-coding with CO/DA affords a tight interweaving of design processes and embodied experiences. Our autoethnographic inquiry shows that this practice requires embracing the complexity and ephemerality of improvisation, navigating through different degrees or control and ambiguity to foster serendipity and emergence. Our improvisation without verbalization favors embodied communication and designs that emerge through a constant negotiation between the performers, through movement, sound and code. Our autobiographical approach and practice contributes to HCI by illustrating how designing in action through improvisation makes space for serendipity, nurturing the sensitivity and flow of embodied experiences.

ACKNOWLEDGMENTS

This research was supported by the CO/DA project (DIM RFSI from Région Ile-de-France) and by the ELEMENT project (ANR-18-CE33-0002) from the French National Research Agency. We want to express our sincere gratitude to Élise Prigent and Lucie Van Nieuwenhuyze for their contributions to the project.

REFERENCES

[1] [n. d.]. Most.js: Monadic Event Stream. https://github.com/mostjs/core/. (Accessed on 09/07/2021).

[2] [n. d.]. vvvv - a multipurpose toolkit | vvvv. https://vvvv.org/. (Accessed on 09/06/2021).

[3] Cycling '74. [n. d.]. Max is software for experimentation and invention. https://cycling74.com/. (Accessed on 04/06/2021).

[4] Samuel Aaron, Dominic Orchard, and Alan F. Blackwell. 2014. Temporal Semantics for a Live Coding Language. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design (FARM '14)*. Association for Computing Machinery, New York, NY, USA, 37–47. https://doi.org/10.1145/2633638.2633648

[5] Sarah Fdili Alaoui and Jean-Marc Matos. 2021. RCO: Investigating Social and Technological Constraints through Interactive Dance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 618, 13 pages. https://doi.org/10.1145/3411764.3445513

[6] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. 2006. OMax Brothers: A Dynamic Topology of Agents for Improvization Learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM '06)*. Association for Computing Machinery, New York, NY, USA, 125–132. https://doi.org/10.1145/1178723.1178742

[7] Engineer Bainomugisha, Andoni Lombide Carreton, Tom van Cutsem, Stijn Mostinckx, and Wolfgang de Meuter. 2013. A Survey on Reactive Programming. *Comput. Surveys* 45, 4, Article 52 (Aug. 2013), 34 pages. https://doi.org/10.1145/2501654.2501666

[8] Steve Benford, Chris Greenhalgh, Andy Crabtree, Martin Flintham, Brendan Walker, Joe Marshall, Boriana Koleva, Stefan Rennick Egglestone, Gabriella Giannachi, Matt Adams, Nick Tandavanitj, and Ju Row Farr. 2013. Performance-Led Research in the Wild. *ACM Trans. Comput.-Hum. Interact.* 20, 3, Article 14 (jul 2013), 22 pages. https://doi.org/10.1145/2491500.2491502

[9] Francisco Bernardo, Michael Zbyszyński, Mick Grierson, and Rebecca Fiebrink. 2020. Designing and Evaluating the Usability of a Machine Learning API for Rapid Prototyping Music Technology. *Frontiers in Artificial Intelligence* 3 (2020), 13. https://doi.org/10.3389/frai.2020.00013

[10] Daniel Bisig and Pablo Palacio. 2016. Neural Narratives: Dance with Virtual Body Extensions. In *Proceedings of the 3rd International Symposium on Movement and Computing* (Thessaloniki, GA, Greece) *(MOCO '16)*. Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/2948910.2948925

[11] Andrew R. Brown and Andrew Sorensen. 2009. Interacting with Generative Music through Live Coding. *Contemporary Music Review* 28, 1 (2009), 17–29. https://doi.org/10.1080/07494460802663991 Publisher: Routledge.

[12] Barry Brown, Stuart Reeves, and Scott Sherwood. 2011. Into the Wild: Challenges and Opportunities for Field Trial Methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1657–1666. https://doi.org/10.1145/1978942.1979185

[13] Diogo Cabral, Urândia Carvalho, Joao Silva, Joao Valente, Carla Fernandes, and Nuno Correia. 2011. Multimodal video annotation for contemporary dance creation. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2293–2298.

[14] Antonio Camurri, Shuji Hashimoto, Matteo Ricchetti, Andrea Ricci, Kenji Suzuki, Riccardo Trocca, and Gualtiero Volpe. 2000. EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems. *Computer Music Journal* 24, 1 (March 2000), 57–69. https://doi.org/10.1162/014892600559182

[15] Antonio Camurri, Gualtiero Volpe, Stefano Piana, Maurizio Mancini, Radoslaw Niewiadomski, Nicola Ferrari, and Corrado Canepa. 2016. The Dancer in the Eye: Towards a Multi-Layered Computational Framework of Qualities in Movement. In *Proceedings of the 3rd International Symposium on Movement and Computing* (Thessaloniki, GA, Greece) *(MOCO '16)*. Association for Computing Machinery, New York, NY, USA, Article 6, 7 pages. https://doi.org/10.1145/2948910.2948927

[16] Yves Candau, Jules Françoise, Sarah Fdili Alaoui, and Thecla Schiphorst. 2017. Cultivating Kinaesthetic Awareness through Interaction. In *Proceedings of the 4th International Symposium on Movement and Computing (MOCO '17)*. ACM, London, UK, 35–40. https://doi.org/10.1145/3077981.3078042

[17] Marianela Ciolfi Felice, Sarah Fdili Alaoui, and Wendy E. Mackay. 2018. Knotation: Exploring and Documenting Choreographic Processes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 448, 12 pages. https://doi.org/10.1145/3173574.3174022

[18] Nick Collins. 2011. Live Coding of Consequence. *Leonardo* 44, 3 (June 2011), 207–211. https://doi.org/10.1162/LEON_a_00164

[19] Nick Collins and Alex McLean. 2014. Algorave: Live Performance of Algorithmic Electronic Dance Music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths, University of London, London, United Kingdom, 355–358. https://doi.org/10.5281/zenodo.1178734

[20] Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. 2003. Live Coding in Laptop Performance. *Organised Sound* 8, 3 (Dec. 2003), 321–330. https://doi.org/10.1017/S135577180300030X

[21] Audrey Desjardins and Aubree Ball. 2018. Revealing Tensions in Autobiographical Design in HCI. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 753–764. https://doi.org/10.1145/3196709.3196781

[22] Audrey Desjardins and Ron Wakkary. 2016. Living In A Prototype: A Reconfigured Space. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5274–5285. https://doi.org/10.1145/2858036.2858261

[23] Carlos Gonzalez Diaz, Phoenix Perry, and Rebecca Fiebrink. 2019. Interactive Machine Learning for More Expressive Game Interactions. In *2019 IEEE Conference on Games (CoG)*. 1–2. https://doi.org/10.1109/CIG.2019.8848007

[24] Carolyn Ellis, Tony E. Adams, and Arthur P. Bochner. 2011. Autoethnography: An Overview. *Historical Social Research / Historische Sozialforschung* 36, 4 (138) (2011), 273–290. http://www.jstor.org/stable/23032294

[25] S Eriksson, A Unander-Scharin, V Trichon, C Unander-Scharin, H Kjellström, and K Höök. 2019. Dancing With Drones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'19)*. ACM, New York, NY, USA.

[26] Sarah Fdili Alaoui. 2019. Making an Interactive Dance Piece: Tensions in Integrating Technology in Art. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) *(DIS '19)*. ACM, New York, NY, USA, 1195–1208. https://doi.org/10.1145/3322276.3322289

[27] Sarah Fdili Alaoui, Baptiste Caramiaux, Marcos Serrano, and Frédéric Bevilacqua. 2012. Movement qualities as interaction modality. In *Proceedings of the Designing Interactive Systems Conference (DIS'12)*. ACM Press, Newcastle, UK, 761. https://doi.org/10.1145/2317956.2318071

[28] Sarah Fdili Alaoui, Jules Françoise, Thecla Schiphorst, Karen Studd, and Frederic Bevilacqua. 2017. Seeing, Sensing and Recognizing Laban Movement Qualities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 12 pages. https://doi.org/10.1145/3025453.3025530

[29] Marianela Ciolfi Felice, Sarah Fdili Alaoui, and Wendy E. Mackay. 2021. Studying Choreographic Collaboration in the Wild. In *Designing Interactive Systems Conference 2021* (Virtual Event, USA) *(DIS '21)*. Association for Computing Machinery, New York, NY, USA, 13 pages. https://doi.org/10.1145/3461778.3462063

[30] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human Model Evaluation in Interactive Supervised Learning. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11*. ACM Press, Vancouver, BC, Canada, 147. https://doi.org/10.1145/1978942.1978965

[31] Jules Françoise, Yves Candau, Sarah Fdili Alaoui, and Thecla Schiphorst. 2017. Designing for Kinesthetic Awareness: Revealing User Experiences Through Second-Person Inquiry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, Denver, CO, USA, 5171–5183. https://doi.org/10.1145/3025453.3025714

[32] Jules Françoise, Norbert Schnell, Riccardo Borghesi, and Frédéric Bevilacqua. 2014. 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, 287–292.

[33] William W. Gaver. 2005. The video window: my life with a ludic system. *Personal and Ubiquitous Computing* 10, 2-3 (aug 2005), 60–65. https://doi.org/10.1007/s00779-005-0002-2

[34] William W Gaver, Jacob Beaver, and Steve Benford. 2003. Ambiguity as a resource for design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 233–240.

[35] Nicholas Gillian, Benjamin Knapp, and Sile O'Modhrain. 2011. A Machine Learning Toolbox for Musician Computer Interaction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Oslo, Norway, 343–348. https://doi.org/10.5281/zenodo.1178031

[36] Nicholas Gillian and Joseph A Paradiso. 2014. The Gesture Recognition Toolkit. *Journal of Machine Learning Research* 15 (2014), 3483–3487.

[37] Andrea Giomi. 2020. Somatic Sonification in Dance Performances. From the Artistic to the Perceptual and Back. In *Proceedings of the 7th International Conference on Movement and Computing* (Jersey City/Virtual, NJ, USA) *(MOCO '20)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/3401956.3404226

[38] Greg Guest, Kathleen MacQueen, and Emily Namey. 2012. *Applied Thematic Analysis*. SAGE Publications, Thousand Oaks, California. https://doi.org/10.4135/9781483384436

[39] Yasamin Heshmat, Carman Neustaedter, and Brendan DeBrincat. 2017. The Autobiographical Design and Long Term Usage of an Always-On Video Recording System for the Home. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) *(DIS '17)*. Association for Computing Machinery, New York, NY, USA, 675–687. https://doi.org/10.1145/3064663.3064759

[40] Sarah Homewood, Amanda Karlsson, and Anna Vallgårda. 2020. Removal as a Method: A Fourth Wave HCI Approach to Understanding the Experience of Self-Tracking. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1779–1791. https://doi.org/10.1145/3357236.3395425

[41] Kristina Höök, Baptiste Caramiaux, Cumhur Erkut, Jodi Forlizzi, Nassrin Hajinejad, Michael Haller, Caroline Hummels, Katherine Isbister, Martin Jonsson, George Khut, Lian Loke, Danielle Lottridge, Patrizia Marti, Edward Melcer, Florian Müller, Marianne Petersen, Thecla Schiphorst, Elena Segura, Anna Ståhl, Dag Svanæs, Jakob Tholander, and Helena Tobiasson. 2018. Embracing First-Person Perspectives in Soma-Based Design. *Informatics* 5, 1 (feb 2018), 8. https://doi.org/10.3390/informatics5010008

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

[43] Stacy Hsueh, Sarah Fdili Alaoui, and Wendy E. Mackay. 2019. Deconstructing Creativity: Non-Linear Processes and Fluid Roles in Contemporary Music and Dance. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 203 (Nov. 2019), 21 pages. https://doi.org/10.1145/3359305

[44] Stacy Hsueh, Sarah Fdili Alaoui, and Wendy E. Mackay. 2019. Understanding Kinaesthetic Creativity in Dance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300741

[45] Caroline Hummels, Kees C. J. Overbeeke, and Sietske Klooster. 2007. Move to get moved: a search for methods, tools and knowledge to design for expressive and rich movement-based interaction. *Personal and Ubiquitous Computing* 11, 8 (Oct. 2007), 677–690. https://doi.org/10.1007/s00779-006-0135-y

[46] Dhruv Jain, Audrey Desjardins, Leah Findlater, and Jon E. Froehlich. 2019. Autoethnography of a Hard of Hearing Traveler. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) *(ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 236–248. https://doi.org/10.1145/3308561.3353800

[47] Alexander Refsum Jensenius and Michael J Lyons. 2017. *A Nime Reader: Fifteen Years of New Interfaces for Musical Expression*. Vol. 3. Springer.

[48] Lian Loke, George Poonkhin Khut, Maggie Slattery, Catherine Truman, Lizzie Muller, and Jonathan Duckworth. 2013. Re-sensitising the body: interactive art and the Feldenkrais method. *International Journal of Arts and Technology* 6, 4 (2013), 339–356.

[49] Lian Loke and Toni Robertson. 2013. Moving and making strange. *ACM Transactions on Computer-Human Interaction* 20, 1 (March 2013), 1–25. https://doi.org/10.1145/2442106.2442113 Place: New York, NY, USA Publisher: ACM.

[50] Andrés Lucero. 2018. Living Without a Mobile Phone: An Autoethnography. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 765–776. https://doi.org/10.1145/3196709.3196731

[51] Andrés Lucero, Audrey Desjardins, Carman Neustaedter, Kristina Höök, Marc Hassenzahl, and Marta E. Cecchinato. 2019. A Sample of One: First-Person Research Methods in HCI. In *Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion* (San Diego, CA, USA) *(DIS '19 Companion)*. Association for Computing Machinery, New York, NY, USA, 385–388. https://doi.org/10.1145/3301019.3319996

[52] Thor Magnusson. 2011. Algorithms as Scores: Coding Live Music. *Leonardo Music Journal* 21 (2011), 19–23. https://doi.org/10.1162/LMJ_a_00056

[53] Thor Magnusson. 2011. Ixi Lang: A SuperCollider Parasite for Live Coding. In *Proceedings of the International Computer Music Conference*.

[54] Thor Magnusson. 2014. Herding Cats: Observing Live Coding in the Wild. *Computer Music Journal* 38, 1 (March 2014), 8–16. https://doi.org/10.1162/COMJ_a_00216

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

[56] James McCartney. 2002. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal* 26, 4 (Dec. 2002), 61–68. https://doi.org/10.1162/014892602320991383

[57] Alex McLean. 2014. Making Programming Languages to Dance to: Live Coding with Tidal. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design (FARM '14)*. Association for Computing Machinery, New York, NY, USA, 63–70. https://doi.org/10.1145/2633638.2633647

[58] Carman Neustaedter and Phoebe Sengers. 2012. Autobiographical design. *Interactions* 19, 6 (nov 2012), 28–33. https://doi.org/10.1145/2377783.2377791

[59] Radoslaw Niewiadomski, Maurizio Mancini, Stefano Piana, Paolo Alborno, Gualtiero Volpe, and Antonio Camurri. 2017. Low-Intrusive Recognition of Expressive Movement Qualities. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI '17)*. Association for Computing Machinery, New York, NY, USA, 230–237. https://doi.org/10.1145/3136755.3136757

[60] Jérôme Nika, Marc Chemillier, and Gérard Assayag. 2017. ImproteK: Introducing Scenarios into Human-Computer Music Improvisation. *Comput. Entertain.* 14, 2, Article 4 (Jan. 2017). https://doi.org/10.1145/3022635

[61] François Pachet, Pierre Roy, Julian Moreira, and Mark d'Inverno. 2013. Reflexive Loopers for Solo Musical Improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2205–2208.

[62] Miller Puckette et al. 1996. Pure Data: Another Integrated Computer Music Environment. *Proceedings of the second intercollege computer music concerts* (1996), 37–41.

[63] Nicolas Rasamimanana, Frédéric Bevilacqua, Norbert Schnell, Emmanuel Fléty, and Bruno Zamborlin. 2011. Modular Musical Objects towards Embodied Control of Digital Music Real Time Musical Interactions. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI'11)*. Funchal, Portugal, 9–12.

[64] Jean-Philippe Rivière, Sarah Fdili Alaoui, Baptiste Caramiaux, and Wendy E Mackay. 2019. Capturing Movement Decomposition to Support Learning and Teaching in Contemporary Dance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 86.

[65] Charlie Roberts, Matthew Wright, JoAnn Kuchera-Morin, and Tobias Höllerer. 2014. Gibber: Abstractions for Creative Multimedia Programming. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia* (Nov. 2014), 67–76. https://doi.org/10.1145/2647868.2654949

[66] Yvonne Rogers and Paul Marshall. 2017. Research in the Wild. *Synthesis Lectures on Human-Centered Informatics* 10, 3 (2017), i–97.

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

[68] Thecla Schiphorst. 2011. Self-evidence: Applying Somatic Connoisseurship to Experience Design. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. ACM, New York, NY, USA, 145–160. https://doi.org/10.1145/1979742.1979640

[69] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi. 2009. Mubu & Friends - Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/Msp. In *Proceedings of International Computer Music Conference*. Montreal.

[70] Sertan Şent́urk, Sang Won Lee, Avinash Sastry, Anosh Daruwalla, and Gil Weinberg. 2012. Crossole: A Gestural Interface for Composition, Improvisation and Performance Using Kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. University of Michigan, Ann Arbor, Michigan. https://doi.org/10.5281/zenodo.1178201

[71] Kate Sicchio. 2014. Hacking Choreography: Dance and Live Coding. *Computer Music Journal* 38, 1 (March 2014), 31–39. https://doi.org/10.1162/COMJ_a_00218

[72] Kate Sicchio and Alex McLean. 2017. Sound Choreography ¡¿ Body Code: Software Deployment and Notational Engagement without Trace. *Contemporary Theatre Review* 27, 3 (2017), 405–410. https://doi.org/10.1080/10486801.2017.1343244

[73] Kate Sicchio, Zeshan Wang, and Marissa Forbes. 2020. Live Coding Tools for Choreography: Creating Terpsicode. In *Proceedings of the 2020 International Conference on Live Coding (ICLC2020)*. University of Limerick, 87–94. https://doi.org/10.5281/zenodo.3939135

[74] Scott Smallwood, Dan Trueman, Perry R. Cook, and Ge Wang. 2008. Composing for Laptop Orchestra. *Computer Music Journal* 32, 1 (March 2008), 9–25. https://doi.org/10.1162/comj.2008.32.1.9

[75] Andrew Sorensen. 2005. Impromptu : An Interactive Programming Environment for Composition and Performance. In *Australasian Computer Music Conference 2009 : Improvise*.

[76] Dan Trueman. 2007. Why a Laptop Orchestra? *Organised Sound* 12, 2 (2007), 171–179. https://doi.org/10.1017/S135577180700180X

[77] Danielle Wilde, Thecla Schiphorst, and Sietske Klooster. 2011. Move to Design/Design to Move: A Conversation About Designing for the Body. *interactions* 18, 4 (July 2011), 22–27. https://doi.org/10.1145/1978822.1978828 Place: New York, NY, USA Publisher: ACM.

[78] Xiao Zhang and Ron Wakkary. 2014. Understanding the Role of Designers' Personal Experiences in Interaction Design Practice. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (Vancouver, BC, Canada) *(DIS '14)*. Association for Computing Machinery, New York, NY, USA, 895–904. https://doi.org/10.1145/2598510.2598556

[79] Qiushi Zhou, Cheng Cheng Chua, Jarrod Knibbe, Jorge Goncalves, and Eduardo Velloso. 2021. Dance and Choreography in HCI: A Two-Decade Retrospective. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 262, 14 pages. https://doi.org/10.1145/3411764.3445804