Title
Preserving Hand-Drawn Qualities in Audiovisual Performance Through Sketch-Based Interaction

Author
Bourgault, Samuelle

Publication Date
2022

University of California
Santa Barbara

Preserving Hand-Drawn Qualities in Audiovisual Performance Through Sketch-Based Interaction

A thesis submitted in partial satisfaction
of the requirements for the degree

Master of Science
in
Media Arts and Technology

by

Samuelle Bourgault

Committee in charge:

Professor Jennifer Jacobs, Chair
Professor Curtis Roads
Professor Yon Visell

December 2022

The Thesis of Samuelle Bourgault is approved.

Professor Curtis Roads

Professor Yon Visell

Professor Jennifer Jacobs, Committee Chair

November 2022

Preserving Hand-Drawn Qualities in Audiovisual Performance Through Sketch-Based Interaction

Acknowledgements

I would like to thank the members of my committee and the MAT faculty members for their insight during the realization of this work.

I want to thank especially my advisor, Professor Jennifer Jacobs, for the academic, financial, and emotional support that she provided during these past three years in MAT. I have grown so much from our conversations and meetings.

I also want to thank my friends and family who make my everyday life worth it.

Finally, I am also extremely grateful for my partner, Jérôme, who is always there to support me, make good food, and comfort me even from afar, and my best friend, Gab, who sends me music recommendations and with whom I have long and insightful discussions about art among many other things.

Abstract

Preserving Hand-Drawn Qualities in Audiovisual Performance Through Sketch-Based Interaction

by

Samuelle Bourgault

Live coding—the real-time procedural creation of audiovisual works—suggests opportunities to extend hand-drawn animation; however, existing live coding systems are incompatible with manual animation workflows. Manual input is not a primary datatype in existing live coding languages and live coding tools require using symbolic programming environments. I theorize that by applying direct manipulation to the domain of live coding, we can enable animators to create expressive mappings between hand-drawn animations and audio effects in real-time. I present Megafauna, a sketch-based system for audiovisual performance, informed by interviews with professional animators. Megafauna supports the integrated generation and control of hand-drawn animation and audio sequences by enabling animators to directly sketch mapping functions between animation frames and sound generators. I demonstrate the expressive potential of Megafauna by reproducing animated compositions from procedural and manual domains. I evaluate the opportunities of our approach for live production through an expert review of a performance piece created with Megafauna.

Contents

Chapter 1

Introduction

Many artists have explored the relationship between audio and visual modalities by creating works that blend sound and images. Artists in the 18th century invented and performed with color organs: devices to display colored light and form with an approach similar to musical performance [1], and contemporary animators and effects artists create animations and visuals for both prerecorded and live musical performance[1]. Many of the foundations of audiovisual expression are rooted in the pre-computational forms of *visual music* [2, 3]. Visual music practitioners approach the creation of visuals with the temporal strategies found in music composition and improvisation. They use a linked set of concepts where visual parameters like dynamics, figures, and shape are considered analogous to musical parameters like rhythm, pitch, phrasing, and timbre [4, 5]. The expressive properties of visual music are especially evident in the work of analog abstract film animators whose work is characterized by hand-drawn visuals (figure: 5.1-a). Creating analog abstract films is labor-intensive and, historically involved the manual synchronization of hand-drawn animation frames with audio. At the same time, manual

[1]e.g. Becky & Jo's music video for Tame Impala (`http://beckyandjoes.com/tame-impala`) and Ryoji Ikeda's Supercodex live set (`http://www.ryojiikeda.com/archive/concerts/#ryoji_ikeda_live_set`)

expression shaped the aesthetics of the work by preserving the style and variation of the human animator.

There are similarities between the practices of analog abstract animators and contemporary digital audiovisual creators. In particular, there are parallels between abstract animation and the field of *live coding*: the performative creation of audiovisual works through computer programming [6]. Similar to abstract animators' use of manual drawing to generate both audio and visual effects, live coders use domain-specific programming languages that create interconnected audio and visual output [7]. Rather than rely on manual synchronization, however, live coders use computational mapping and automation to generate visual effects with audio data or audio effects with visual data. As a result, performers can rapidly generate and iterate on audiovisual compositions in real time [8].

In observing the parallels between abstract animation and live coding, I see opportunities to support improvisation and exploration in manually created audiovisual work by harnessing live coding strategies for procedural generation and synchronization. Furthermore, I theorize that adapting the procedural automation from live coding in a form compatible with manual expression could enable the use of manual animation in live performance.

At present, live coding systems are powerful but poorly aligned with manual animation workflows. Live coding *languages* centers on the manipulation of procedurally generated sound and visual data [9] in contrast to manual drawing, sketching, and painting. Live coding *environments*, like general-purpose programming systems, require editing a representational description of the work. As a result, they limit artists' ability to work through the direct interactions present in physical or analog media workflows [10]. I compared the exploration and automation afforded by live coding with the limitations of live coding systems for manual expression to arrive at my research questions: 1) *How can* *a system support manual artists in authoring dynamic audiovisual works?* 2) *How can drawing enable expressive control of digital audio synthesis?* , and 3) *What opportunities does procedural direct manipulation offer for improvisational live performance?*

I theorize that by applying procedural direct manipulation [11] to the domain of live coding, we can enable animators to use manual drawing to procedurally generate and synchronize images and sound. Direct manipulation can enable creators to express declarative computational relationships in forms that are similar to declarative programming languages [12]. Thus far, procedural direct manipulation techniques have primarily been used to create dynamic illustrations [13] or animation [14]. I seek to extend the application of procedural direct manipulation to audiovisual creation.More specifically I envision a system that enables the live creation of abstract animation films in a form analogous to live coding.

To explore my first and second research questions, I interviewed animators who work across programming, live performance, and direct manipulation. The interviews showed how animators combine manual and automated techniques to control the spatial and temporal properties of hand-drawn animations, and how animators use music and musical concepts to develop visual properties of their work. I used interview themes to develop a direct manipulation animation system for performance that supports the composition and synchronization of hand-drawn animations through drawing and sketching. My system supports performative audiovisual animation by automatically synchronizing hand-drawn animated frames to real-time audio synthesis. My system also supports direct manipulation of audiovisual mappings by enabling animators to describe audio and visual mapping functions by drawing on parameter graphs.

To investigate my third research question, I analyzed eight recent computational audiovisual performances. I used this analysis to define a design space composed of five dimensions. I developed an improvisational performance using my system that corresponds to unexplored points within this design space and conducted an expert review with two prolific audiovisual artists to evaluate the resulting piece.

In this thesis, I make the following contributions. **First**, I present a set of design principles for supporting live audiovisual performance through direct manipulation, derived from interviews with animators. **Second**, I apply these principles in an interactive system entitled *Megafauna*– a sketch-based tool for creating audiovisual works.[2] Megafauna is distinguished from prior work on procedural direct manipulation because it can produce works that combine animated visuals with sound. My approach also offers greater expressiveness in comparison to direct manipulation audiovisual interactive systems with predefined mappings [1, 15, 16] because it enables creators to author custom mapping functions. **Third**, I demonstrate the applications of direct manipulation-based multimodal expression by using Megafauna to recreate existing analog, digital, and procedurally animated works. These applications show how my approach supports rapid creation, exploration, and improvisation in audiovisual compositions while preserving manual visual aesthetics. I analyze the process of applying Megafauna to audiovisual composition to reflect on my design goals and provide recommendations for future multimodal tools that support manual expression. **Fourth**, I explore Megafauna's capabilities to support live performance by using the system to design and execute a complete audiovisual performance. Applying my system to a comprehensive production workflow enabled me to investigate approaches for designing interfaces and representations for live improvisational work and articulate key dimensions of a design space for computational audiovisual performance. I use insights from the expert review of the completed performance to discuss how design decisions for audiovisual technologies may support different opportunities for performance and expression within this design space.

[2] The name Megafauna is a poetic choice. In a performance setting, I imagined creating large projected animated patterns to remember and honor extinct giants. When spoken, the name also shares sonic similarities to the word "megaphone", which seemed appropriate for an audiovisual tool.

1.1 Attributions

The content of chapters 1 to 6 is the result of a collaboration with professor Jennifer Jacobs and has previously appeared in the Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2021 [17]. This thesis extends this VL/HCC submission on developing a sketch-based system for audiovisual performance. I provide an analysis of nine audiovisual performance works and define a five-axis design space for audiovisual performances (Section 7.1). I also contribute a revised version of the Megafauna system, informed by the application of the existing system to the design and execution of a 13-minute-long live performance. My analysis of the performance workflow emphasized the importance of features that optimize readability, and the advantage of audio-visual mapping strategies that produce high degrees of perceivable complexity through low performer effort (Section 7.3). Through the performance process, I extend the Megafauna system with new features including an allocated region of the screen for the control graphs to avoid the overlay of visual information and preset instruments for the performer to rapidly select from and modify. Further, I contribute a discussion of system design considerations for supporting manual input for live performance through the analysis of the creation process of the performance made with Megafauna (Section 8) and through a formal design critique with two experts in animation and performance (Section 9 and 10). In these new sections, I highlight the limitations of my system such as the pressure of live manual content creation and the expectations of narrative structures from frame-by-frame figurative animations. Moreover, I emphasized the opportunities supported by features acting on timelines, computational constraints that preserve expressiveness, and automation of audiovisual synchronization.

Chapter 2

Related Work

My work builds on methods in visual music, live coding, and direct manipulation of dynamic images and sound.

2.1 Visual Music and Abstract Animation

Visual music has origins in early avant-garde cinema and abstract painting of the 1920s [4]. Visual musicians applied temporal strategies from music composition to animation and considered visual properties like rhythm, dynamics, figures, and shape as analogous to musical parameters like rhythm, pitch, phrasing, and timbre [5]. Manual visual musicians created hand-drawn animations with synchronized changes in form, color, and sound. The aesthetic opportunities of manual visual music are particularly notable in the work of abstract animators like Norman McLaren, Mary Ellen Bute, Oskar Fischinger, Lillian Schwartz, and Hans Richter [5, 18]. Analog abstract animation was labor-intensive and could involve taking a picture of each painted frame or drawing directly on the film strip. In some of his most famous pieces,[1] McLaren drew both the

[1] e.g. McLaren's abstract films *Dots* (`https://www.nfb.ca/film/dots/`) and *Loops* (`https://www.nfb.ca/film/loops/`).

visual and the audio content directly on two different filmstrips that he would later align visually and combined into a final filmstrip [19]. This process created audiovisual works that reflected McLaren's drawing style. In developing Megafauna, I was directly inspired by the hand-drawn aesthetics of analog abstract animation and motivated to support these properties in live audiovisual performances.

2.2 Live Coding Practice and Live Coding Systems

Live coding is a heterogeneous performing arts movement that involves generating audio and visuals by writing algorithmic instructions in real time [20]. Performers incorporate elements of improvisation into their performance rather than playing from a predefined score [21], and the opportunity for spontaneity is a key motivation for participants [22]. Live coders perform with symbolic programming systems that support live creation through terse syntax and close domain mapping between code and audiovisual output [21]. Live coding environments support on-the-fly programming activities that allow programmers to write and execute a program while it is running [23], and some environments support networked performance [7, 24]. Strategies in live coding programming languages range from synchronizing events through a timing mechanism in the program flow [6]; representing musical patterns as functions from time to events [21]; and enabling performers to compose object-oriented digital instruments [25]. Collectively, live coding languages emphasize building a regular pattern through a small set of operations and enable rapid high-level changes by altering a small number of variables. Megafauna builds on this approach by creating automatic synchronization of audio and hand-drawn visuals and enabling performers to specify custom audiovisual mapping functions. Live coding performers can also work through visual [26, 27, 28] and tangible [29, 30] programming systems. These systems often use dataflow representations where programs are structured as directed graphs. Visual languages can reduce syntax and composition challenges [31], but they still require artists to work through abstract descriptions [10]. Unlike symbolic textual and visual live coding languages, Megafauna enables artists to author concrete hand-drawn artwork through a graphical direct manipulation interface.

The synchronization between the audio and visuals in a multimedia performance setting— often embodied by a two-person team composed of a musician and a visual artist— also presents some challenges. In these settings, there is often higher importance granted to the music at the expense of the visuals and a lack of conceptual and technical communication between the performers [32]. It's also challenging for performers to develop complex and flexible mappings [33, 34]. By combining audio and visual modalities in one system, Megafauna tries to facilitate their integration by providing a way to generate sound and animations simultaneously.

2.3 Direct Manipulation for Sonic and Visual Expression

Artists and developers have created direct manipulation tools for audiovisual expression. Sutherland developed Sketchpad, the first system that allowed users to interact with computers through drawing supporting both artistic and technical applications [35]. Levin presented a series of "painterly" audiovisual synthesis systems that enabled animators to create animated imagery and sound through drawing [1]. Singing Fingers enables artists to create interactive drawings with brush weight and color mapped to microphone input [15]. Sounding Brush enables artists to use predefined digital brushes that provide inputs to audio-synthesis patches [16]. Acid Sketching enables artists to draw symbols on a physical piece of paper to generate sonic sequences [36] and SoundCodr pushed this idea further by enabling performers to associate hand-drawn shapes to a custom audio grammar [37].These systems demonstrate the expressiveness of drawing for audiovisual creation; however, the majority require artists to use predefined audiovisual mappings. My focus in developing Megafauna was to support artists in authoring custom mappings while retaining a direct manipulation interface.

Researchers and developers have also explored using direct manipulation interfaces to express musical relationships. Applications like SoundBow [38], Different Strokes [39], and AV ZONES [40] enable composers to create synthesizers and sequencers through graphical notation and manually drawn waveforms. These tools have a different objective from Megafauna because their graphical interfaces can only control audio rather than create audiovisual output. Direct manipulation can also be used to define visual procedural relationships that are normally expressed through symbolic programming languages. Victor [41, 42] and Kazi [14] demonstrated sample systems with applications in interactive animation visualization, and Para [11], Sketch-n-Sketch [13] and Palimpsest [43] supported the procedural creation of vector and raster graphics through direct manipulation of constraints, data collections, and duplication behaviors. I extend the approach of procedural direct manipulation in animation and illustration to audiovisual production.

Chapter 3

System Need-finding and Design Goals

To inform the design of Megafauna, I built on my analysis of live coding tools and direct manipulation through interviews with three professional animators who used manual, procedural, and performative approaches. I spoke with performer and manual animator Miwa Matreyek, animator and technologist James Patterson, and software engineer and media artist Kurt Kaminski[1]. I analyzed interview transcripts to distill themes on performance, manual expression, and organization.

3.1 Artist Work Context

Miwa works primarily with pre-rendered animated videos that she creates in Adobe After Effects, a software designed for the production of visual effects and motion graphics. Using the principles of shadow puppetry, she interacts with her animations, creating complex worlds where her shadow takes different narrative roles. Miwa develops precise

[1] Miwa Matreyek: `http://www.semihemisphere.com`, James Patterson: `https://presstube.com/hello`, and Kurt Kaminski: `https://www.kurtkaminski.com`

choreography to synchronize her movements to her videos, which produces embodied performances. In 2020, she won the Golden Nica award for Computer Animation at Ars Electronica for her piece *Infinitely Yours*, a very prestigious recognition in the field of new media arts.

James is an illustrator and animator who develops his own software. His highly unique drawing style conveys whimsical and textural scenes through humor. He works both with two-dimensional and tri-dimensional productions, which enabled him to collaborate on projects ranging from music videos and 2D games to interactive installations and VR applications. With Google Creative Lab, he developed *Norman*, a VR tool to build animations in space. As he builds tools, he iteratively transitions between developing features within the software and creating artwork with the tool. This integration of software development and artwork production provides insight into subsequent features.

Kurt is a Lead Technical Artist at Snap Inc. He worked in the movie animation industry for many years in companies like DreamWorks Animation and Disney Animation. As an artist, he is interested in generative and interactive systems to create immersive experiences. He specializes in the animation of large numbers of components, such as particle systems, using graphic shaders in software that support interactive interactions such as Max/MSP and Unity.

3.1.1 Opportunities of Performance in Animation

Each animator described the benefits of approaching animation as performance. As a performance artist, Miwa created pre-rendered animation segments before live shows and practiced synchronizing her movements to them. As a result, she developed workflows that enabled her to rapidly switch between authoring and performing to get feedback that could inform the design of her animations. James was not a performer, but performance-oriented tools appealed to him. He felt that "perform[ing] animation" could reduce labor and accelerate output. James considered immediacy in software interaction to be critical for maintaining creative flow, stating that "[If] you're more interested in motion through time, then you want this auto advancement. You want [the frame] to advance on the lifting of the pen." Kurt also described how the immediacy of performance-oriented systems was essential to his workflow. He drew parallels from music. Discussing the immediacy of a piano, he said that "without that immediate feedback, it's really hard to create expressive things." These accounts suggested that performance-oriented tools could create multiple opportunities for animators. Speed of execution and immediate feedback are critical properties for live tools. These qualities could also reduce tedium and improve workflows in non-live contexts. Furthermore, tools that enable rapid transitions from authoring to performing could aid animators in experimenting with alternatives prior to a performance.

3.1.2 Expressive Properties of Direct Manipulation

All of the interviewees found direct manipulation interfaces appealing. As an experienced programmer, James wrote a custom software tool that enabled him to author 3D animation by sketching[2]. He described his motivation for doing so as an effort "to make [animation] as physical as possible." Miwa worked exclusively with direct manipulation. She described how direct manipulation was essential because it allowed her "to have [her] hands more in the process". Kurt also described how a combination of direct manipulation and programming could support exploration and variation in his own work. He felt that using manual input to define a sound envelope could allow for richer sonic expression. Overall interviewees considered direct manipulation to be an approachable means of controlling digital behavior for animated and audiovisual works, and a way to

[2]Norman: https://normanvr.com

preserve variation and personal style.

3.1.3 Organizing Audio and Visuals Across Time and Space

The interviewees used a variety of strategies to organize audio and visual elements. James used woven loops: a manual animation technique that involves animating another series of drawings on top of an existing loop. He described creating polyrhythms through woven loops with "variable track lengths" that "come in and out of sync". Miwa organized the visual content of her animation with the music, which acted as the "backbone" of her work. She used markers in Adobe After Effects to annotate the beats of the music and aligned these moments with rough keyframes. To synchronize her animation and music, Miwa used After Effects Time Remapping to dynamically adjust the rate of animation change between two keyframes. Kurt focused on precisely controlling audiovisual synchronization. He noted that working with visual and audio content created different challenges. It was easier for him to identify flaws in a complex audio composition; however, inconsistencies in an animated visual composition were "less obvious" and harder to address. These accounts demonstrate that animators structure visual elements of their work with respect to audio content and that some animators think about manipulating images in ways that are analogous to audio rhythmic structures. Kurt's experience also suggests that while aesthetic and structural concepts like rhythm may be salient for both audio and visual authoring, perceptual differences between visual and audio modalities can require different approaches for organization and review.

James and Miwa also described methods for organizing multiple visual elements in spatial and temporal patterns. Miwa used automated methods to duplicate animations in combination with manual adjustments. She described how she manually animated five different versions of plant growth patterns, multiplied them through digital copypaste, and manually adjusted the spatial and temporal position of each copy. Manual placement ensured the result didn't "feel too computer generated." James combined procedural duplication with hand-drawn animations, similar to a particle system. He procedurally duplicated a small subset of hand-drawn animations and varied their scale, rotation, and position. To ensure the hand-drawn animations could be arbitrarily generated, he composed the action in the frames so that they spawned in the first frame, and decayed in the last frame. This enabled him to instantiate multiple copies in an arbitrary fashion without breaking the illusion of motion. James and Miwa both benefited from automated duplication, however, their approach to controlling duplicates highlighted key trade-offs between manual and procedural organization strategies. Manual manipulation enabled precise control of the synchronization of duplicated animations, whereas procedural control enabled rapid exploration and experimentation while enforcing tighter constraints on the structure of each duplicate.

3.1.4 System Design Goals

I used the interview themes and my analysis of audiovisual works and tools to generate my design goals for a direct manipulation audiovisual authoring system:

1) *Support for hand-drawn expression:* Artists should be able to generate hand-drawn animations in their own style and synchronize hand-drawn animation with audio effects.

2) *Ability to specify audiovisual mappings through direct manipulation:* Artists should be able to define the timing of audiovisual events and control the parameters of audiovisual effects through direct manipulation.

3) *Support for improvisation:* Artists should be able to engage in rapid editing, exploration, and iteration when creating audiovisual content at a speed that is compatible with live performance.

Chapter 4

Megafauna System Description

Megafauna is a stylus-based application for authoring audiovisual animations. I developed Megafauna through an iterative design process. I created sample applications with hard-coded audiovisual mappings. I then analyzed conventions in direct manipulation tools and live coding languages to inform the development of two additional prototype systems with flexible interactions and mapping models. I applied these prototypes to the creation of sample audiovisual works to converge on the final prototype.[1]

4.1 Interface and Implementation

Megafauna is an iPad Pro application built with Swift using the iOS AudioKit framework [44]. Artists use a stylus to author animation frames and specify mapping functions. Megafauna contains five different modes of interaction which can be toggled by icons at the bottom of the canvas (figure 4.1). In these modes, artists can manipulate three core primitives: audiovisual sequences, timelines, and mapping functions.

[1]See the video figure: https://vimeo.com/573196681

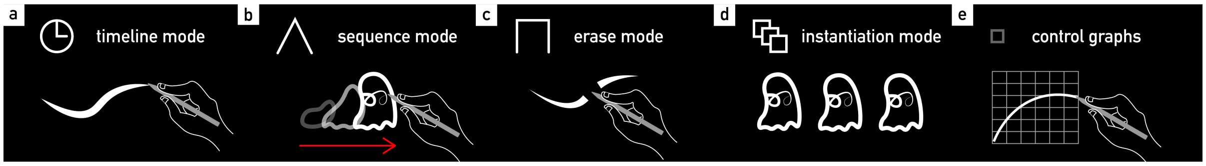

Figure 4.1: Modes of interaction with Megafauna's interface.

4.2 Audiovisual Model

I used a sequencer approach for Megafauna's audiovisual model. The system enables artists to manipulate the properties of discrete, sequential audiovisual events in a manner similar to the way a musician adjusts the pitch and duration of notes triggered by a musical sequencer. Megafauna's authoring process is structured around creating *sequences*: linear sequences of hand-drawn frames of animation. Each sequence frame is automatically paired with an audio oscillator state which consists of a frequency-modulated sine wave. The oscillator state is triggered when the corresponding frame of the sequence is displayed. Each sequence is therefore a combination of audio and visual events. This constraint is imposed in order to support synchronization between animation and synthesized audio. The playback of each sequence is managed by a *timeline*: a temporal loop represented by a hand-drawn stroke. The duration of the loop is determined by the length of the stroke (figure 4.2-a). Each timeline has its own playhead that progresses through the loop at a fixed speed. Sequences are ordered on a timeline and triggered to begin when this playhead passes their location (figure 4.2-c). Megafauna supports the creation of multiple timelines of varying duration. The timeline mode in the Megafauna interface (figure 4.1-a) allows artists to draw new timelines and add audiovisual sequences to the current timeline. The erase mode (figure 4.1-c) enables artists to mute/hide a specific sequence on the current timeline. By defining timelines through drawing, artists can quickly specify multiple loops with varying durations and create polyrhythmic patterns. This enables effects similar to when live coders create emergent melodies by initializing multiple looping sequencers of varying lengths.

Sequence audiovisual mapping behavior in Megafauna is determined by a set of polynomial functions that control the visual properties of the drawing and parameters of the oscillator at each frame of the sequence. Artists specify the mapping function for each property by sketching a pattern on a two-axis control graph (figure 4.3). Table 4.1 lists the full range of control graph audiovisual properties. Control graphs are displayed in the Megafauna interface as an overlay over the current animation when the artist enters control graph mode (figure 4.1-e). To support additional audiovisual complexity, sequences can be procedurally duplicated through the instantiation mode (figure 4.1-d). Each duplicated sequence results in the "duplication" of the corresponding oscillator via additive synthesis. The properties of duplicated sequences can be modified individually in control graph mode, by selecting the individual duplicate's glyph or collectively, by selecting the group glyph (figure 4.4).

Table 4.1: Spatial and temporal properties for audiovisual mappings

	Spatial	Temporal		Spatial	Temporal
Visual	position rotation scale opacity	timing frame order frame rate	Audio	panning	frequency amplitude reverb delay

4.3 Megafauna Workflow

An artist begins in Megafauna in the *timeline* mode, by drawing a timeline of their desired duration. Once the timeline is drawn, the speed and location of the timeline are shown by a playhead visualization that progresses along the drawn line (figure 4.2-c). The artist can then add a sequence (figure 4.2-a) by double tapping on the timeline to indicate when the sequence should be triggered. Megafauna displays a cursor on the timeline to help the artist select their desired location. The artist then enters the *sequence* mode and can begin drawing their animation frame by frame (figure 4.2-b). A gray dot will appear on the timeline to indicate the location of the sequence. Megafauna restricts sequence frames to one continuous stroke. Each time the artist lifts the stylus from the screen, the system will progress to the next frame. This enables rapid animation while constraining individual frames to simple forms. To end the sequence, the artist clicks on the *sequence* mode button. From then on, each time the timeline playhead passes on the specified location, the sequence will be triggered (figure 4.2-c). By default, the oscillator for each sequence is set to C_0—a low-frequency pitch that avoids disturbing the ongoing audiovisual piece. To modify the sound parameters, frame order, and frame transition, the artist can access the sequence control graphs by clicking on the sequence glyph—a small grey square located next to the first vertex of the first frame of the animation (figure 4.3-a). In *control graph* mode, the artist is presented with a GUI that shows a set of sketchable graphs labeled with the corresponding audio or visual parameters (figure 4.3-b).

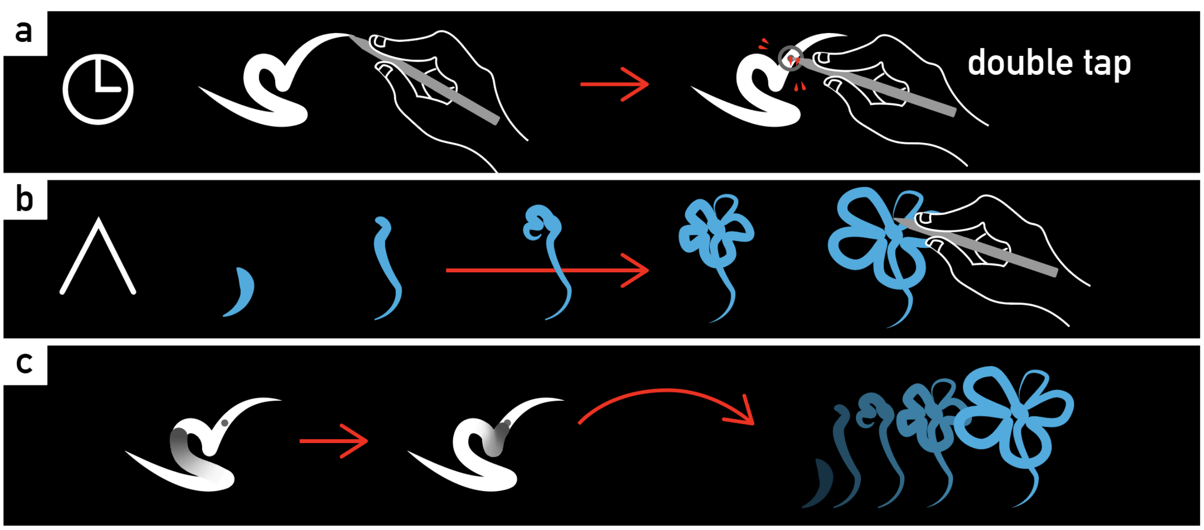

Figure 4.2: Making a timeline and adding an audiovisual sequence to it. a) Drawing the timeline and double tapping on a location along it to define the temporal position of the sequence that gets represented by a grey dot. b) Drawing the sequence frame by frame. c) The result: the sequence will be triggered when the playhead passes the dot representing that sequence.

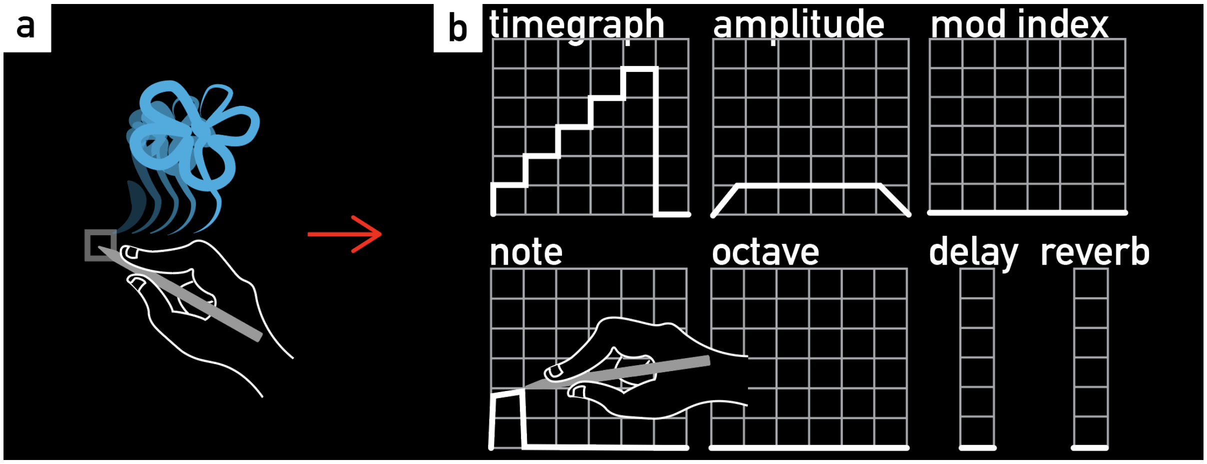

Figure 4.3: Controlling the audiovisual parameters over time using control graphs. a) The glyph associated with a sequence is a button to access the control graphs GUI. b) The artist can draw their own functions to define these mappings.

The artist can procedurally duplicate a sequence by entering *instantiation* mode (figure 4.1-d), and selecting the sequence they wish to duplicate by double tapping the sequence icon on the timeline. Sequence instantiation uses an interaction similar to the automatic progression of frames in *sequence* mode. Artists automatically progress through the instantiation workflow through pen-up gestures. Once a sequence is selected the artist can specify the number of copies of the sequence, and their positions by drawing. A new sequence will be instantiated at the first vertex of the stroke they make on the canvas. Lifting and then drawing a new stroke will instantiate another copy (fig: 4.4). The artist can repeat this workflow to add additional sequences to the same timeline, create additional timelines, or further modify the parameters or instantiation of their existing sequences. The artist can also stop a specific sequence to be triggered—meaning the animation will be hidden and the audio will be muted—by selecting *erase* mode and then using the stylus to erase sections of the timeline where sequence markers are located. Artists can restart the sequence by re-drawing the erased section. This allows for rapid editing of the loop.

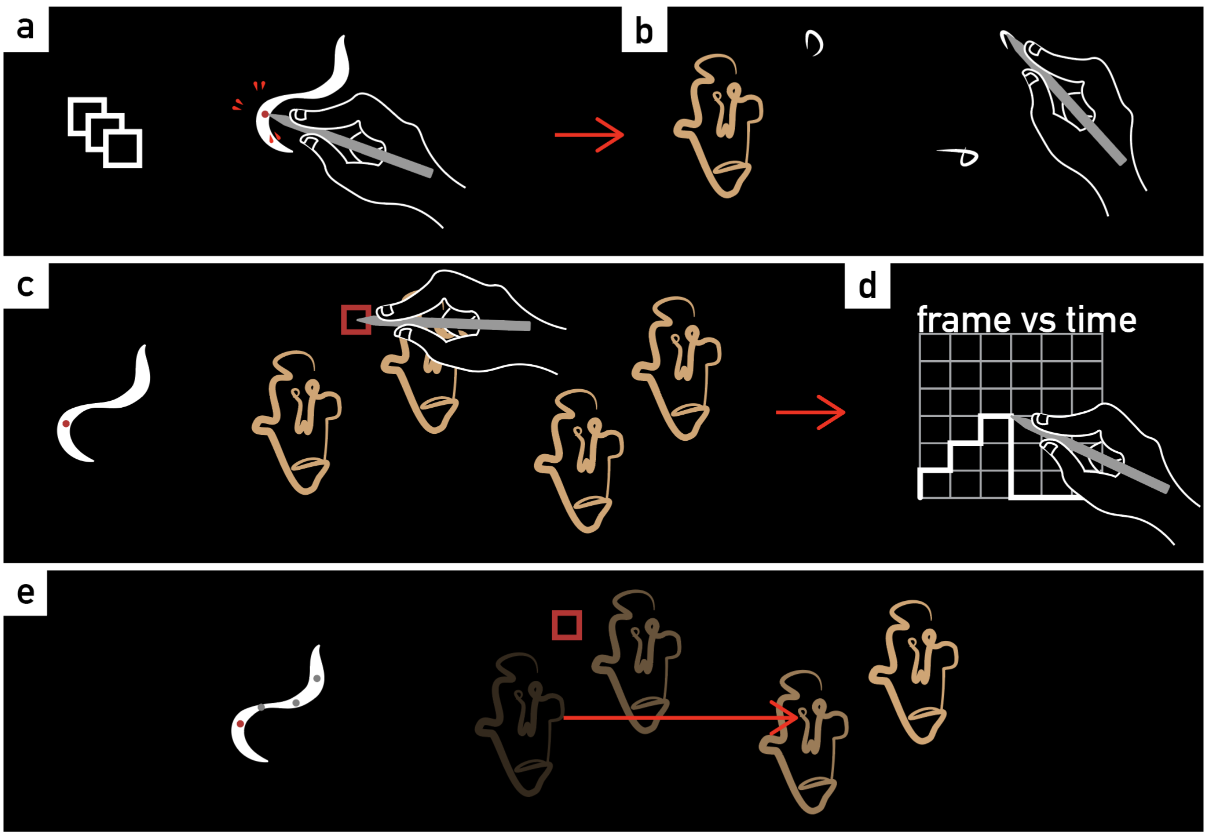

Figure 4.4: Sequence instantiation. a) Selecting a sequence to instantiate on the timeline. b) Drawing a sequence of positions for the instances. c) The instances. d) Controlling the instances' location in time via the group control graphs. e) Dots on the timeline indicating each instance's location in time.

Chapter 5

Demonstrative Evaluation

I evaluated the expressive range of my approach through a demonstrative application of Megafauna to animations and performative works. I present three example applications created entirely with my system: 1) recreating an analog audiovisual abstract animation, 2) augmenting silent hand-drawn animations with audio effects, and 3) real-time improvisation of an audiovisual work with hand-drawn aesthetics.

5.1 Example 1: Rapidly Creating Manual Abstract Animation

In this example, I demonstrate how my approach enables the rapid recreation of an analog abstract film while preserving the hand-drawn characteristics of the original piece. I recreated an excerpt of the McLaren film *Dots*. This piece is characteristic of the genre and was made by manually drawing animation frames and sound markers on the film strip. The first 10 seconds of the piece include eight dots that sequentially appear and shrink in a semicircle path. A flower-like shape shrinking in size follows the animation of the dots. Short audio effects punctuate the appearance of each shape (figure: 5.1-a).

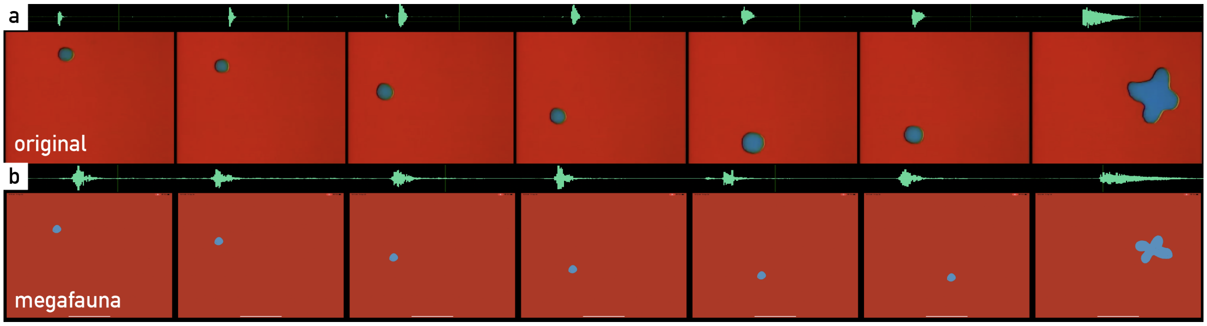

Figure 5.1: Example 1: Reproducing an excerpt of Norman McLaren's piece *Dots*. a) The original soundtrack and frames. b) The version made using Megafauna.

I began in Megafauna by drawing one animated dot sequence that I accelerated slightly using the time control graph. I added a "bleep" sound by drawing a short amplitude envelope and increasing the modular index of the FM oscillator in the control graph. I then instantiated that sequence seven times and positioned the copies around a semicircle. I used the group graph to define the interval of time between each of them. I finished by drawing the last flower-like shape on the timeline after the representation of the last instantiated dot (figure: 5.1-b). The entire process took three minutes.

5.2 Example 2: Adding Audio to Duplicated Manual Animation

In the second example, I recreate a silent animation made with manual and procedural approaches, and I automatically generate and synchronize sound with the visuals. I reproduced the silent animation of a group of ghosts as they appear in my interviewee James Paterson's video entitled *A 25 minute silent ambient animation compilation*[1]. The animation is structured around a small number of simple hand-drawn loops which are duplicated and positioned at different intervals in time and space to create a visual

[1]See: `https://vimeo.com/53259598`

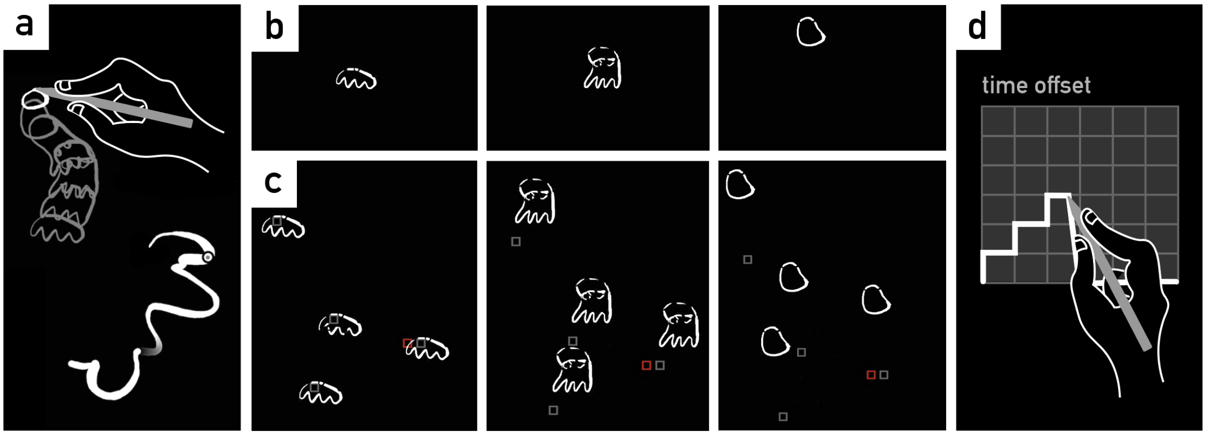

Figure 5.2: Example 2: Making James Paterson's group of ghosts. a) Drawing each frame. b) The frames of the ghost sequence being triggered one after the other in time. c) The instantiation of four copies animated in time. d) Using the offset graph to relocate the instances in time.

polyrhythm (figure 5.3-a).

I began in Megafauna, by generating a single sequence with a ghost animation that appeared in the first frame and disappeared in the last frame (figure 5.2-a,b). I used the *instance* mode to duplicate the ghost and varied the start time of each instance by using the group temporal control graph (figure 5.2-c). Because each copy of the animated ghost automatically has an associated FM oscillator, instantiating multiple copies automatically produced a corresponding additive soundtrack. To create variation in the audio, I modified the pitch of the first ghost sequence to C_1 in the pitch control graph. I then used the individual control graph for each instance to change the octave to create additional audio variation between copies (figure 5.2-d). This re-creation (figure 5.3-b) took three minutes to create. This example shows the potential of automatic audiovisual synchronization and procedural duplication for hand-drawn animation. Furthermore, it demonstrates how Megafauna enables live audiovisual production without the use of a symbolic programming language.

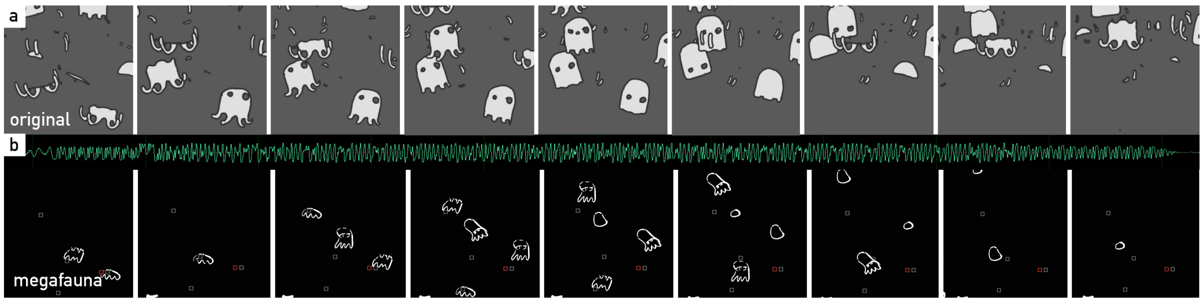

Figure 5.3: Example 2: The final result. a) The original animation. b) The animation made with Megafauna and its associated soundtrack.

5.3 Example 3: Live Performance and Improvisation

My final example explores the performance potential of Megafauna by applying it to the creation of an improvisational work (figure 5.4). This example takes advantage of two functions that were not used in the previous examples: (1) drawing multiple timelines and (2) erasing and redrawing parts of the timelines to respectively stop and start the triggers for audiovisual events on that portion of the timeline. The first function–multiple timelines–enabled me as the performer to generate polyrhythmic effects in both the visual and the audio domain, a strategy often used in live coding to create complexity. The second function– manually editing the timeline behavior through drawing– facilitated the creation of visual and sonic dynamics during the improvisation/performance session. I created multiple abstract animated sequences along multiple timelines. After doing so, I found that I could use drawing interactions to quickly move from a very dense audiovisual landscape to a lighter state with only a few sequences left unmuted. If I wished to restore the muted sequences back, I could quickly and easily re-draw the missing timeline segments. The complete improvisational performance took five minutes to compose and edit.

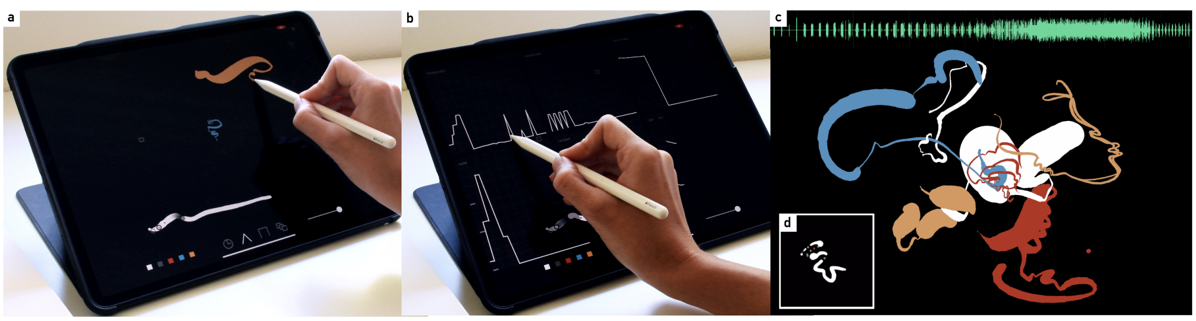

Figure 5.4: Megafauna, a direct manipulation tool to author audiovisual artwork. a) Hand-drawing a visual frame. b) Controlling sonic parameters in a sketch-based graphical interface. c) Performance piece made with the system. d) A timeline with some erased sections to *mute* audiovisual events.

Chapter 6

System Discussion

I discuss the outcomes of my approach by examining the demonstrative applications with respect to my original design goals: (1) supporting manual expressiveness, (2) controlling audiovisual mappings through direct manipulation, and (3) supporting improvisational work. I compare Megafauna's expressiveness to other direct manipulation audiovisual systems and textual live coding tools and describe tensions between manual and procedural creation that emerged in my use of the system.

6.1 Comparing Megafauna to Direct Manipulation AV Systems

Megafauna achieves my first and second design goals, in part, through automated audiovisual synchronization and an interface that supports both frame-based animation and the sketching of mapping functions. My example applications demonstrate how Megafauna's audiovisual model and direct manipulation interface enables greater aesthetic range and functionality in comparison to prior examples of direct manipulation tools for audiovisual authoring [1, 15].

I compare Megafauna to six prominent drawing-based audiovisual tools to discuss this point (see Table 6.1).

Rather than restrict creators to predefined mapping parameters between audio and visual elements, Megafauna enables the creation of sequences for which the only explicit link between the audio and the visual events is their synchronization. Artists can freely alter the parameters of the audio and visual components of the sequence while ensuring synchronization is maintained. This structure enables the artists to deviate from preset mappings and define their own relationships between hand-drawn frames and FM sounds. For instance, one animator could want the animation of a shrinking circle to correspond to a linear increase in frequency while another animator could associate this visual sequence with a quadratic decrease in frequency. This flexibility is important because audiovisual mappings are subjective and different creators favor different qualities [45].

Megafauna is differentiated from prior audiovisual direct manipulation tools in part because it supports the creation of animated drawings that are unbound from the audio timeline. In the six systems that I analyzed in Table 6.1, each drawing acts as a trigger for the audio, meaning that the expressiveness of the drawings is limited by the audio composition requirements or the spatial constraints of the audio representation. Furthermore, in most cases, the drawing serves as static visual input used to generate dynamic audio. Out of the audiovisual direct manipulation tools I surveyed, only *Yellowtail* and *Sonic Wire Sculptor* support the creation of animation, but these animations are constrained spatially within the audio frequency space. In both tools, a line has to be drawn in a specific location on the screen to produce a specific pitch. Tools that are not bound by space such as *Singer Fingers* and *SoundCodr* allow the real-time specification of custom mapping between sound and visual but do not support animation. Megafauna's integration of user-specified audiovisual mappings and decoupling of animation and audio timelines supports a unique procedural output resembling the aesthetics of abstract analog animated films.

Table 6.1: Comparison of Drawing-based Audiovisual tools. Megafauna differs from these prominent tools by the type of visuals it can produce and its direct-manipulation specification of custom mappings between sound and visuals.

Tool	Type of Visual	Mappings with Sound
Yellowtail [1]	Animated drawing as trigger	Sound properties (pitch) mapped to line and spatial properties
Acid Sketch [36]	Static drawing as trigger	Sound properties (pitch, envelop, resonance, etc) mapped to shape properties (area, roundness, orientation, etc)
Singing Fingers [15]	Static drawing as timeline	Drawings can be mapped to any voice-recorded sound
Sonic Wire Sculptor [46]	3D animated drawing as sequencer	Sound properties (trigger, pitch, panning) mapped to line spatial properties
SoundCodr [37]	Static drawing as trigger	Drawing can be mapped to any sound through visual programming
SoundBow [38]	Static drawing as timeline	Sound properties (pitch, rhythm) mapped to line spatial properties and drawing speed
Megafauna	Frame-by-frame animated drawings and static drawing as timeline	Sound and visual properties specified in control graphs

6.2 Comparing Megafauna to Textual Live Coding Tools

To further examine the success of my system with respect to my second and third design goals, I discuss how Megafauna compares to textual live coding systems in terms of flexibility and live creation. My examples show how manual control can support aspects of textual live coding without symbolic programming syntax. To illustrate this point, I compare Megafauna to two prominent symbolic live coding languages. In the audiovisual language Gibber, the creation of a simple drum pattern can be written by $a = Drums('xoxo')$, where the symbol x corresponds to a kick drum hit and the symbol o to a snare drum hit [7]. In ORCA, MIDI events are triggered by alphanumeric character methods that are organized along a two-dimensional grid. One example method is the delay method D, which sends a *bang* every beat. If the output of the function is aligned with a MIDI message such as : $03C$, where 0 is the first MIDI channel, 3 the third octave, and C is the note, that message will be triggered [47].

These examples demonstrate two common properties of live coding languages. First, regardless of language, live coding patterning algorithms focus on rapid alteration rather than enforcing a rigid order of audio events. This is powerful because manipulating the order of audio events can produce generative outcomes that would be difficult through manual methods. Second, despite adopting similar procedural methods, textual live coding languages contain different syntax and modus operandi. As a result, similar to learning different general-purpose programming languages, learning a new framework requires learning a new syntax. Because Megafauna is direct manipulation based, it does not have textual syntax. Megafauna is not as computationally expressive as a textual language; however, it preserves the ability to generate and modify audiovisual patterns in real-time and experimentally adjust algorithmic parameters in a manner similar to textual languages. In the process, Megafauna eliminates the requirement to learn textual programming syntax. At this stage, I cannot make strong claims on the ease of learning Megafauna; however, past work demonstrating the ease of use of direct manipulation in comparison to textual programming [11] suggests that my approach has promise for broadening participation in live audiovisual creation. Future user studies with Megafauna could provide additional insights into how easy the system is to learn.

6.3 Tensions between Manual and Procedural Creation for Exploration and Improvisation

My third example demonstrates how Megafauna supports exploration and improvisation in line with my third design goal. The process of improvising with two modalities in Megafauna revealed key tensions in combining procedural audio with manual animation. As discussed in the previous section, live coding algorithms enable the flexible ordering of sound events. By contrast, frame-based animations require rigid ordering. While changing the order or number of musical notes in a phrase can be generative, modifying the order or the number of visual frames in an animation will break the illusion of motion. Frame-based animation also requires rigidity in the timing and duration of visual frames. In Megafauna, an irregular frame rate can be interesting in the context of experimental work, as shown in Example 3, when the frames of a sequence are not visually following one another; however, a constant frame rate is required to maintain the persistence of vision for more traditional sequential animations (e.g. Example 1 and 2). By comparison, in many audio sequences, note duration can be adjusted with greater flexibility without an undesirable perceptual effect. Some variations between different notes can even contribute an analog quality to the sequence.

I found that the differences between procedurally generated audio and frame-based animation shaped how I used control graphs to control animations and sound in Megafauna. I needed more precision when manipulating the control graph for visual frames. For instance, the graph controlling the timing of the frames is set to a step function by default, where the x-axis corresponds to time and the y-axis to the frames. To modify the frame rate of an animation, the artist would need to systematically redraw the step function while precisely resizing it across the time axis (the y-axis). Such care is not as critical when manipulating the control graphs associated with audio effects. These observations suggest that systems aiming to integrate manual artwork and audio should support different kinds of exploration and variation for sequences of sound and sequences of visuals. Many symbolic live coding languages use generic audiovisual data structures and representations to facilitate mapping. By contrast, my evaluation of Megafauna suggests that when integrating manual animation and procedural audio, the desired perceptual properties of both mediums should inform the representations and constraints on manipulation for synchronized audiovisual objects.

Chapter 7

Investigating Procedural-Manual Methods for Live Audiovisual Performance

I investigated my third research question – *What opportunities does procedural direct manipulation offer for live audiovisual performance?* by applying Megafauna to the design and execution of an improvisational performance. I spent two weeks using Megafauna to conceive of and execute a completed performative work. During the day, I tested different audiovisual approaches with Megafauna in a multi-media room on the University of California Santa Barbara campus. In the evening, I modified the Megafauna system functionality by adding or adjusting features in response to our experimentation.

Developing a performance with Megafauna had several advantages. (1) It allowed me to evaluate the performance of my tool in extended use for the creation of a polished and complex creative work. (2) It enabled me to define a design space applicable to the audiovisual performance domain that both highlights opportunities for new audiovisual systems research and provides a mechanism for evaluating the output of audiovisual performance tools based on the aesthetic properties they support. (3) It required me to further refine the system interface and representations in response to the requirements of my performance. In doing so, I gained broader insights into system design methods to support rapid manual drawing for live performance. The process also exposed new challenges not previously encountered through my demonstrative evaluation.

My approach was informed by prior design research methodology. I draw from research through design (RtD) methods in which the design of artifacts supports the investigation of research domains with ill-defined and complex problems [48]. My approach also shares features with autobiographical design methodologies wherein researchers build the system, use it themselves, learn about the design space, and evaluate and iterate the design based on their own experiences [49]. Autobiographical design methods are particularly applicable for research contributions where novelty resides in the integration of many technical research contributions from a variety of disciplines into a single working system [48]. My focus on creativity support tools for multimodal live creation necessitates that I evaluate the resulting technology through practical applications while providing opportunities to reflect on the multiple dimensions of performance. I augment the investigations of Megafauna through personal design practice with an external expert review of the resulting artifacts. I describe this review further in Section 9.

In the remainder of this section, I describe the design space that informed the development of the Megafauna performance. I follow by detailing the ideation and execution of the performance itself.

7.1 Design Space for Manual Audiovisual Performance

To inform the development of the Megafauna performance and the expert evaluation, I created a design space for audiovisual performance. I selected prominent examples of current audiovisual performances in the field of new media art. Many of these performances were developed by technologies that I describe in Section 2.2. I analyzed these performances with respect to my original design goals for Megafauna in Section 3.1.4. From this process, I conceptualized five design dimensions that oriented the design choices for the Megafauna performance. To help illustrate these dimensions, in figure 7.1j, I contrast eight example audiovisual works (Fig 7.1a to 7.1h) and the performance made with Megafauna (Fig 7.1i) based on their positioning across each dimension axis. Each work is categorized from the perspective of a particular performance piece made with an audiovisual system rather than all the performative possibilities available by that system. Below, I detail each dimension by providing examples of how different performances' aesthetic properties map to different points along each axis.

Creation Process: I observed that different performances provided varying degrees of transparency to aid the audience understand the decisions and actions of the performer. I mapped sample works with respect to their degree of *transparency* or *opacity*. I characterized live coding performances like Alexandra Cárdenas' and Olivia Jack's live set at ICLC 2019 (Fig 7.1e) as having a transparent creation process because the audience is able to see lines of code being written and edited in real-time. This practice also demonstrates the scripting virtuosity of the performers. By contrast, performances like Ryoji Ikeda live set (Fig 7.1b) and Rhizomatics drone choreography (Fig 7.1c) are opaque because the audience is not able to view the executed actions of the performer to create audiovisual content. I modeled Megafauna as a live coding technology, and thus, I sought to preserve the live coding tradition of transparency in my performance. To achieve this, I integrated the authoring interface as part of the projected visuals. The performer was also allowed to toggle between the authoring mode and the presentation mode where the UI elements disappeared leaving only the animated sequences. My intention was to provide flexibility to the performer to select which elements they wanted the audience to

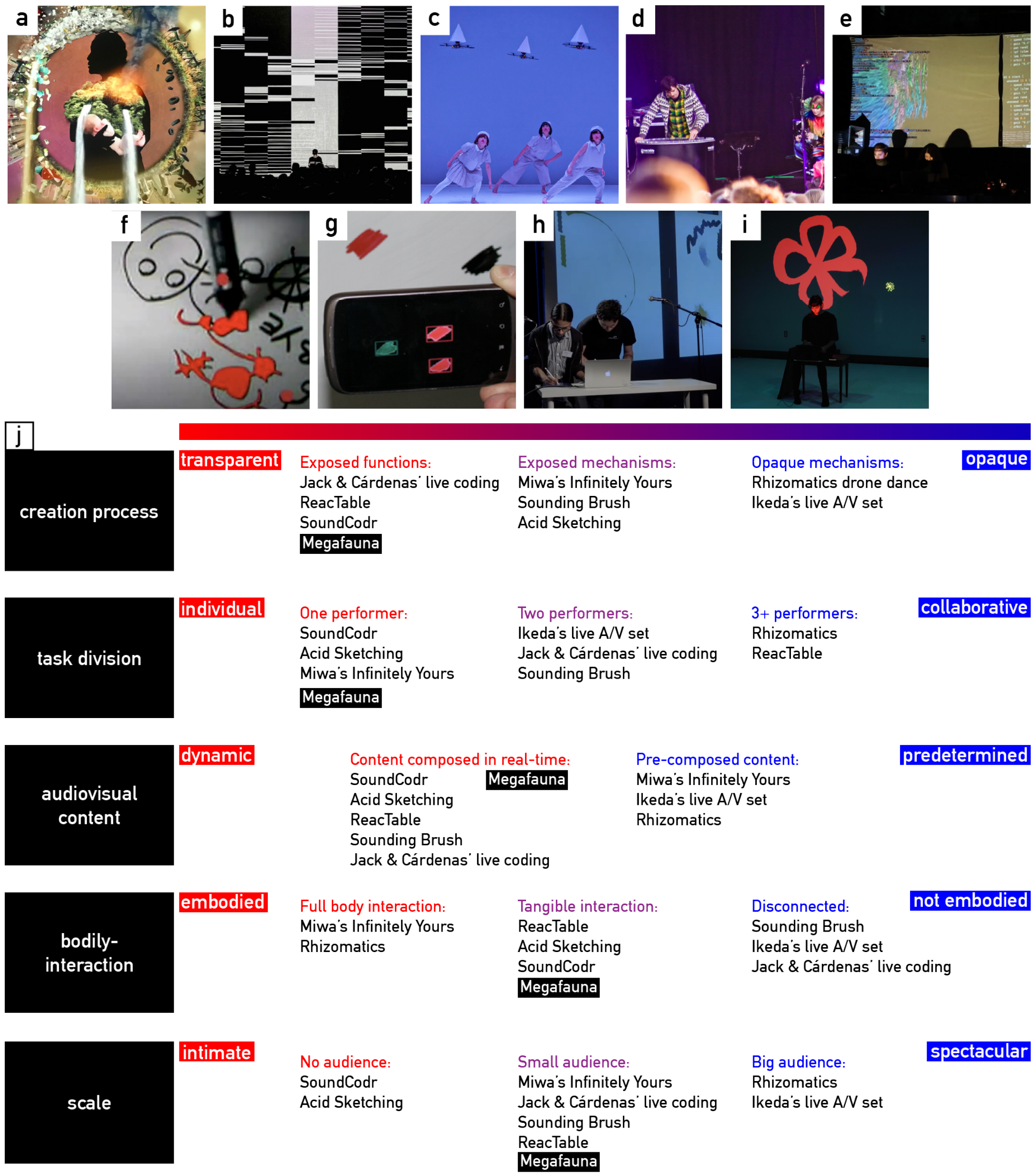

Figure 7.1: Design Space for Audiovisual Performance. a) Miwa's Infinitely Yours. b) Ryoji Ikeda's live set. c) ELEVENPLAY x Rhizomatics drone and dance performance. d) The reacTable by Sergi Jordà, Günter Geiger, Martin Kaltenbrunner and Marcos Alonso used in Björk concert. e) Alexandra Cárdenas and Olivia Jack live coding performance at ICLC 2019. f) Performance demo with Acid Sketching by Alex McLean. g) Performance demo with SoundCodr by Alistair Stead, Alan Blackwell and Samuel Aaron. h) Sounding Brush by Sourya Sen, Koray Tahiroğlu and Julia Lohmann. i) Megafauna performance piece. j) Example works organized along five design space dimensions.

focus on: the authoring process or the creation itself.

Task Division: Live coding and more traditional audiovisual performances are often *collaborative*, where the tasks of visual creation and sound production are divided among two or more performers. This task division enables the performers to focus on one modality at a time. By contrast, some performers have focused on work that can be performed *individually*. I found that this quality was preserved in systems such as Acid Sketching (Fig 7.1f) and SoundCodr (Fig 7.1g). I developed the Megafauna performance as a solo piece, with all audio and visuals executed by one individual. This approach enabled me to test the effectiveness of the Megafauna system to support artists working primarily with visual animation tools to design their own sound.

Audiovisual Content: Live coding is also characterized by *dynamic* content, where audiovisual effects are generated in real-time during a performance. Dynamic approaches to live creation allow for improvisation and novelty in each performance instance. In contrast, other forms of audiovisual performance (e.g. Fig 7.1a and 7.1c) are generally prerecorded or choreographed. Predetermined performances reduce risk, and allow for a high level of precision and repeatability, but lack the variability of dynamic performance. With Megafauna, I wanted to create an improvisational performance. I developed a piece in which all animations were drawn during the performance. I relied on the automatic audio mapping feature to enable the performer to focus their labor on manual drawing. This approach ensured that, even if the performer recreated the same drawings, each performance would be unique, because the drawings, and subsequent audio effects, would be different.

Bodily-Interaction: Live coding and other audiovisual performances often follow a laptop-to-screen model, where the performer is performing from their laptop and is physically disconnected from the performance [50, 9]. In forms of *embodied* performance, like dance, performers' bodies are a component of the spectacle witnessed by the audience and kinesthetic movement plays a role in creative ideation and exploration [51]. ELEVENPLAY x Rhizomatics (Fig 7.1c) work demonstrates the integration of computational effects and physical dance by having dancers coordinate their actions to the movement of drones' movements, while in performances made with the reacTable (Fig 7.1d), part of the spectacle involves performers physically manipulating tangible objects to generate sound. Embodied interactions break the common division of space between the performers and the projection space and enable the creation of a hybrid space where audiovisual content and performers' bodies coexist and communicate with one another. To achieve this effect with Megafauna, I decided to have the performer present in the projection space (Fig 7.1i), a decision that was not planned for in the system development and only arose during the creation process of the performance. This configuration enabled me to create moments where animated sequences were projected directly on the performer's body – their face, stomach, and hand. The Megafauna system was aimed at preserving elements of embodiment by reflecting the visual drawing style of the performer using the system. I hypothesized that integrating on-body projection and manual drawing would help to convey a physical and embodied aesthetic in my performance.

Scale: Audiovisual performances vary greatly in scale. I characterize performances like Ryoji Ikeda's live set (Fig 7.1b) or ELEVENPLAY x Rhizomatics dance show (Fig 7.1c) as *spectacular* in that they aim to be presented to a large audience. I observed that performances created by systems like Acid Sketching (Fig 7.1f), SoundCodr (Fig 7.1g), and Sounding Brush (Fig 7.1h) are conducive to *intimate* performances with small groups or even individual viewers. The aesthetics of a small-scale performance structure appealed to me, because of the potential it offered for the performer to connect directly with the audience members. The decision to use the performer's body as a projection surface was, in part, related to the desire to create an intimate environment. This approach limited the dimensions of the space to a human scale and required the audience to be physically closer to the performer in order to see the visuals. Moreover, enabling real-time authoring of audiovisual sequences placed the performer vulnerable position, which I hoped would foster emotional engagement from the audience.

Overall, the design choices made during the development of the Megafauna performance represent one particular area in the larger space of audiovisual performance; I sought to support live coding practices of transparency and dynamism while integrating notions of embodiment and intimacy that are more common in manual audiovisual performances.

7.2 Inspiration for the performance

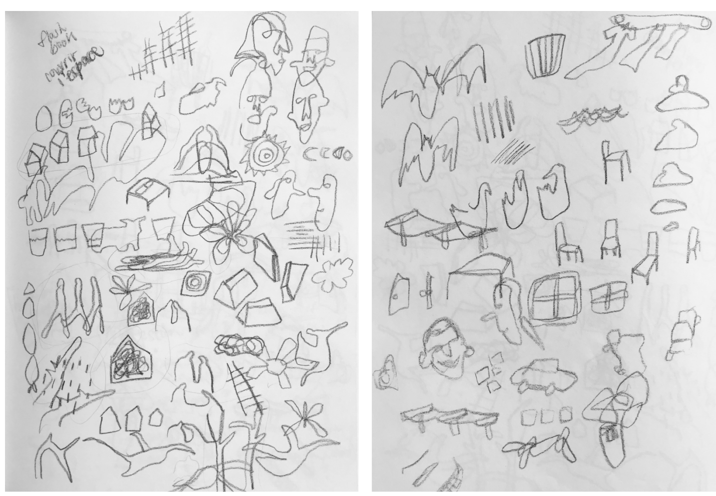

Figure 7.2: The performer's sketchbook pages when preparing the Megafauna performance.

To create this performance, I was interested in building on the hand-drawing capabilities of Megafauna to create a performance that evoked a "sketchbook coming to life". As seen in Figure 7.2, artists' sketchbooks are often a collection of seemingly unrelated drawings that cohabit on the same page. These often dense juxtapositions of drawings create a unique aesthetic that I wished to extend to the projection space. In my performance

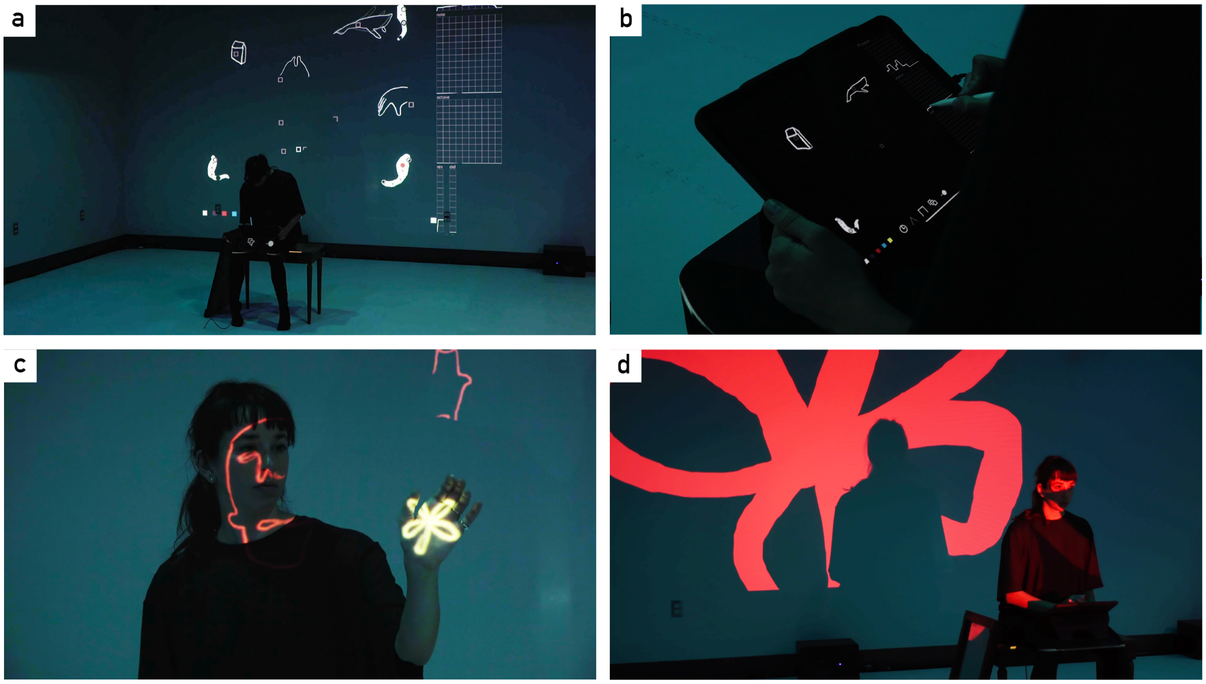

Figure 7.3: Documentation of the Megafauna performance. a) Full view of the performance space. b) The Megafauna system used during the performance. c) Embodied interactions between the performer's body and the hand-drawn animated sequences. d) Final scene of the performance when a red flower blows up.

titled *(un)cover* (or *(dé)couvrir* in French, which means both uncover and discover), the performer covers the space and their own body with simple animations, while creating a loop-based soundscape. This series of animated drawings presents a loose narrative on the thematic of the house and the self that currently inhabits the performer's thoughts, the house being a place for self-reflection that has been overwhelmingly prominent in the past years of the global pandemic. I also decided to include elements of embodied interaction between the performer's body and the animated sequences in the performance (see Fig 7.3c). In doing so, I wished to convey a self-reflective narrative by connecting illustrations directly to parts of the performer's body. Documentation images of the performance created with Megafauna are shown in Figure 7.3 and the video documentation is available at https://vimeo.com/679885026.

7.3 System revisions to facilitate live performance

One of the challenges of the dynamic and individual performance piece was to create enough content to keep the audience engaged without overwhelming the performer. During the design process, I observed that features that optimize higher complexity for lower effort helped support the needs of real-time audiovisual performance.

To add more complexity to the performance while maintaining a manageable amount of options, I defined three instruments that the performer can choose from: an FM oscillator (present in the previous version of the tool), an additive synthesizer comprised of three sine wave oscillators of different frequencies, and a pink noise generator. The performer uses radio buttons visible on the authoring interface to select an instrument. Each audiovisual sequence is then associated with the instrument selected on the radio buttons at its creation. For instance, if the pink noise generator is selected when creating the sequence, the sequence will produce a pink noise sound when it is triggered on the timeline. Adding instruments enabled a wider range of sonic qualities that allowed for more specificity in the audio and visual relationship. I wanted to create a rain animation during the performance, for example, and needed a percussive sound to associate with each drop so the pink noise generator was an ideal choice.

In line with live-coding practices and as a way to present a transparent workflow, I updated the authoring interface to respect my performance need. In live coding, text is generally overlaid on computationally generated visuals. In the previous interface design, I overlaid the control graphs on the canvas, the drawing space. In a performance context, however, such an interface hid the audiovisual sequences, which felt distracting. To overcome this issue, I added a scrollable sidebar on the right side of the screen to access the control graphs but had to reduce the space allocated to drawing to accommodate this new interface component.

Chapter 8

Observations from making a performance with Megafauna

My creation process leading to the making of an improvisational performance piece helped me surface opportunities and limitations of a procedural direct manipulation system used in a live context. In this section, I present three considerations: the pressure that live manual content puts on the performer, the way features that act on the timeline lower the workload for the performer, and how certain constraints on audiovisual parameters restrain the performance in useful ways.

8.1 Live manual content creation puts pressure on performer

Live audiovisual performance puts a lot of pressure on the performer. They must create a significant amount of content in real-time while dealing with the possibilities of system failures or lack of inspiration [22]. In performing with Megafauna, I found that stress on the performer arises from two elements: (1) repeatedly performing actions that require manual skill and (2) dealing with multi-modal (visual, audio, and embodied) creation. Due to the manual skills required for drawing each animation frame, I observed that the visual content was more demanding to perform than the audio generation, which only involves manipulation through a graph interface. The modalities at play – composing sound, drawing, and planning for embodied interactions with the visual components – also increase the level of concentration required by the performer during the live creation process. Time spent on drawing implies that the performer is not spending time crafting sounds and vice versa. Additionally, I observed that features requiring a specific order of execution also needed more focus on the performer's part. For instance, my system lets a performer define the instruments associated with a sequence only at the creation of that sequence. This implies that the performer must know the instruments they want to generate before starting to draw the sequence. If they are unhappy with their decision they cannot easily reverse it. This constraint adds another layer of elements for the performer to be aware of before executing an action.

8.2 Features acting on timeline facilitate live performance

I noticed that features that act on timing properties while avoiding the creation of new content support live performance workflow by lowering the effort required from the performer and increasing the complexity of the audiovisual content. For instance, when making instances of a sequence, the group control graph enables the rapid positioning of the copies along the timeline through one-stroke movement. Through minimal effort, the performer can therefore quickly increase the density of audiovisual content. The mute and unmute feature enabled by erasing and redrawing the timelines also supports rapid changes in the sound and visual scapes. With simple gestures, the performer can turn content that had already been created earlier in the performance on and off. By contrast, I found that changes on individual sequence is effort intensive and time-consuming for little payback. Currently, most of the performance is spent on tasks like drawing frames or modifying sound parameters one by one.

8.3 Some computational constraints are useful for live performance

During the making of the Megafauna performance, I identified constraints that help define the boundaries of the real-time creation process. These constraints allow orienting the performer's focus on creating varied outputs rather than making decisions on what elements to modulate next. For instance, each frame of an audiovisual sequence is comprised of a single line, so each time the performer lifts their stylus, they automatically jump to the next frame. This feature has two benefits for constraining the performance space. First, it forces the performer to draw quick and simple illustrations and avoid them spending too much time on complicated designs. Second, it generates a distinctive aesthetic that contrasts with traditional animation tools.

Limiting the amount of available sonic parameters and instruments also offers constraints that help lower the cognitive load on the performer. It circumscribes the range of possible options for the performer to control while maintaining variability in the outcomes. Parameters that affect the frequency of the generated sounds such as the modular index of the FM synthesis generator can be changed with a simple gesture in the control graph and return a wide variety of sound effects.

In my experience of performing with Megafauna, I felt like I could constrain the audiovisual available options while maintaining a wide range of expressive outcomes.

Chapter 9

Expert Evaluation

To evaluate the performance that I created with Megafauna and orient future development of the system, I conducted an expert review with Miwa Matreyek and James Paterson, two of the animators that I interviewed previously in the need-finding phase of the system. I chose to conduct an expert review to obtain detailed feedback on the artifact created with the Megafauna system. My approach builds on prior systems research heuristic evaluation [52]. By defining a set of design objectives that corresponded with opportunity spaces within audiovisual production, and then evaluating the adherence of my performance to those design objectives through expert review, I sought to gain insight into the broader opportunities of integrating manual drawing and audio production for live performance, and articulate opportunities for future audiovisual systems research.

I selected James and Miwa as evaluators because of their expertise across illustration, programming, animation, and live performance. Both are prolific artists and are actively recognized by the art and technology community. Therefore their insights were precise and extremely valuable for defining the limitations and opportunities of my system. The expert review took place one week after completing the Megafauna performance. I sent both reviewers a document containing evaluation questions to consider in their critique.

These questions were organized under three main topics that aligned with the audiovisual design space: the aesthetic of the sound and the visuals, the aesthetic of the authoring interface, and the aesthetic of the interaction between the projected elements, the space, and the performer's body. I instructed James and Miwa to review the questions, watch a 6-minute video demonstrating the Megafauna system, and watch the documentation of my 13-minute performance, in that order. Following a 3-day period to review the materials, I conducted an hour-long individual zoom discussion with James and Miwa. I recorded and transcribed the discussions, and reviewed the resulting data. In the remainder of this section, I report on James' and Miwa's evaluation of the Megafauna performance across three themes: the aesthetic of embodied and audiovisual components, the tensions between the authoring interface and the audiovisual creation, and the role of storytelling and narrative emergence in real-time hand-drawn performance.

9.1 Critique of embodied and audiovisual aesthetic

Miwa and James, both, found the embodied interactions between the performer's body and the visual sequences to be the most successful aspect of the performance. James mentioned how "This was like where [he] was like slipping into like a wow." Miwa appreciated the real-time mapping happening on instances when the performer was "composing [sequences] to be on certain surfaces, like her hand." For her, these moments felt like the performer's body became part of the visual outcome, blending "into the images". She also saw the potential for live mapping onto additional spatial objects that could "become activated as like cityscapes with like a whole world of like, you know, cars and airplanes and people walking as like these short animation loops."

Both experts also found the synchronization between audio and visuals successful. James mentioned that "[he] was going back and forth between paying attention to the sound, [and] paying attention to the audio where one was capturing [his] attention more primarily than the other." The experience was different for Miwa. While she understood that there was a relationship between the audio and visuals, she felt the visuals were more dominant because the audience could directly observe the performer authoring them with the stylus. In her words, "we [saw] the process of it being made."

Despite a "undeniable connection" between the audiovisual events, James found the general quality of the piece to induce discomfort. While acknowledging that this is a valid aesthetic, he said the outcome is "not about slickness and smoothness and gentleness. The colors themselves are... interrelated in a very harsh way." James suggested that some forms of automation in the audiovisual creation process could smooth out the outcomes. He gave the example of "one drawing [that] like erupts into life, both [in] sound and animation." Similarly, Miwa wondered if underlying structures could help to "turn [generated sound clips] into song or add something underneath." For instance, she envisioned a way to have the generated tones change keys to create melodies. Miwa also found that the visuals lacked movements and wished for the sequences to move in space. "Because of the shortness of the animation, the number of frames, um, there was a way that they felt like gifs to me."

9.2 Tensions between the authoring interface and the audiovisual creation

Both Miwa and James identified tensions between the audiovisual creation and the real-time authoring process. James felt the Megafauna authoring interface was interesting when presented as a demonstration of the tool but distracting as part of the performance. He described how seeing the authoring interface "was popping [him] out of" the performance and was “breaking the spell” of “the actual final product or the artifact that the system creates.”

James described how the moments during the performance when the interface was turned off felt like a “relief”, and at those stages, he could begin enjoying the artwork. He added that he didn’t need the presence of the interface to understand that the piece was authored in real-time. Rather, seeing the performer work on their iPad was sufficient to convey this live quality. James also described how the points at which the performer was engaged in authoring (e.g. drawing new animation on the tablet) created important moments of “silence” in the piece. He compared these phases to forms of silence in other media– for example, he described how musicians use literal moments of silence to delineate audio compositions, while visual artists rely on negative space to construct compelling visual compositions. For him, silences in live-authoring performances act as moments of “suspense” that keep the audience’s attention on the performance He felt the Megafauna performance would be improved by having more silent moments to break up the otherwise complex phases of animation and sound.

In contrast to James, Miwa felt that seeing the creation process through the performer’s actions on the Megafauna interface was an important part of the piece “to really get the sense of like the real-time manipulation.” Miwa felt however that some elements of the interface were less disruptive to the other elements of the performance, while other components provided a constant distraction from the animation. For instance, she thought that the hand-drawn timelines were a successful part of the UI. Comparing them with traditional straight timelines as found in audiovisual software like Adobe After Effects, she described Megafauna timelines as whimsical and beautiful. She saw also the control graphs as obvious elements of the user interface that can appear and disappear based on the performer’s needs. She mentioned that seeing the performer scrolling through these graphs gave her “a sense that they’re more temporary”. Other elements, however, like the toolbox at the bottom of the projection and the glyphs associated with each sequence seemed more permanent, “we spent so much time in that space that you just kind of start to think that this is the presentation mode.” She believed that these elements should not be as prevalent if the goal is to distinguish the presentation mode from the authoring mode.

9.3 Role of storytelling and narrative emergence in live audiovisual performance

Although it was not my intention to create a narrative work with Megafauna, both experts described the Megafauna performance as a form of storytelling.

James used two different metaphors to describe the feelings the performance evoked. He described the work as both 1) a sketchbook being created in front of your eyes and 2) a collection of drawings reminiscent of early human cave paintings. It was important for him to make sense of the performance narrative through these analogies. James also felt that the piece conveyed strong narrative elements when the performer drew on their own body. He described these moments as a “self-referential autobiographical space,”. The placement of figurative drawings on the performer led him to wonder about the underlying narrative told through these body-animation interactions.

Miwa also found narrative qualities in the performance. When referring to the figurative animation sequences, including the birds, the house, and the face, she stated: “once you start to have ... these elements that look like something, I think the audience starts seeking a reason why they’re there, their relationship to each other, and what the world is that’s being constructed.” Though these narrative elements were present in the artwork, she didn’t feel that they were pointing toward a clear narrative or what she referred to as "world-building." For her, seeing the animation disappearing and then reappearing later on felt like "they're accumulating, but they're not really like creating a world together."

Miwa recommended two strategies to augment the piece through world-building. First, she suggested using a motion path to animate the position of the sequences, an approach she uses in direct manipulation software like Adobe After Effects. For Miwa, giving animation trajectories could help with forging a narrative. Second, she suggested using representational components to generate textural ones, using the "swarm" as a metaphor to describe this effect. She noted that the Ghost example presented in the demonstrative evaluation section was an interesting occurrence of a texture being generated by multiple agents, but that this feature could have been used more in the performance piece.

Chapter 10

Implications for designing computational performance technologies

Based on my observations during the creation process of the Megafauna performance and the expert review of that performance, I discuss implications for designing procedural direct manipulation systems for live performance. I reflect on 1) forms of automation that support or impede virtuosity, 2) the expectations of narrative structures on systems that enable figurative manual drawing, and 3) the opportunity for embodied interactions in real-time authoring performance.

10.1 Forms of automation can support virtuosity in direct manipulation system for performance

Automation increase production efficiency in creative systems, however, automation alone does not necessarily lead to an increase in expressive potential [53]. The expressive potential of a system depends, in part, on the range of outcomes it can support. For example, in visual arts, an expressive software tool must enable the creation of works that display the unique style of the artist who is using it or afford various levels of mastery [54].I argue, therefore, that there is an opportunity to consider how expressive computational systems can incorporate automation as a means to complement artist virtuosity, rather than replace it.

In live-coding performances, performers demonstrate virtuosity, in large part, through the skillful manipulation of symbolic languages to generate sounds and visuals. Live-coding performers must overcome the challenges posed by the real-time production of audiovisual content that conforms to a specific and generally strict syntax. However, symbolic languages provide the possibility to specify both the audiovisual content created and its organization in time. As discussed in section 6.2, a textual command like $a = Drums('xoxo')$ in a live-coding language like Gibber indicates the type of sound (kick drum for x and snare hit for o) as well as their sequencing in time. By contrast, in a direct manipulation tool like Megafauna, expressiveness comes from the act of drawing. Drawing enables the rapid creation of visual frames but requires additional scaffolding to specify the organization of these frames across time. In Megafauna, I automated the progression of the sequence of frames and the triggering of audio effects on each frame to overcome this limitation. Both experts perceived the effect of this automation positively, saying that sound and visuals felt connected to one another.

In the early stages of this project, I looked at multiple audiovisual applications (Section 6.1) and noted that most of them involved constraints that took away the possibility of generating unique outcomes. They, for instance, imposed a set of predefined sound mappings and offered only generic audiovisual effects. In the revised system targeted to live performance, I focused on automating features that would reduce the effort performers needed to expend on synchronizing audio and visuals, while largely avoiding automation in the generation of the visuals themselves. Through the control graphs, I also provided the possibility for the performer to define custom audio mappings for each visual sequence and the modulation in time of these mappings. Through an RtD process, I realized that features affecting the timing of audiovisual sequences relieved the performer from engaging in the structural task of manually specifying the distribution of audiovisual events across time but did not affect the visual quality of the drawings. These temporal features included erasing and redrawing the timeline and organizing multiple instances across time at once. James and Miwa suggested other strategies for automation during the expert review. James suggested using a bank of pre-recorded sequences that could be dragged and dropped on the canvas, or developing generative animation effects that could emerge from a single hand-drawn frame. Miwa suggested defining motion paths for the sequences and developing the instance duplication feature to create textures and control swarms of animated objects that were automatically moved through space in relation to the motion path. These forms of automation could potentially support my goal of expressiveness while avoiding overly constraining manual drawing virtuosity and the diverse aesthetics that emerge from different artists who draw by hand.

My evaluation of the Megafauna system through the making of a performance and an expert review points toward trade-offs for automation in performance tools that involve manual drawing. It can reduce the flexibility to create sound parameters in real-time but can help artists focus more on the manual creation of visuals rather than the manual arrangement of events across time. Considering the role of automation in direct-manipulation systems for live audiovisual performance is important. Different approaches will shape how the performer develops skills with the system and how the authoring interface is developed to support specific objectives.

10.2 Real-time creation of manual drawing is better suited for figurative content but adds expectations of a narrative structure

The majority of live coding performances I reviewed for this paper featured complex geometric visuals. This is primarily due to the fact that the logical and mathematics primitives used in symbolic procedural image generation often lend themselves more easily to the creation of abstract visuals rather than figurative ones [55]. Although it is possible to draw both abstract and figurative content in Megafauna– as evidenced in my demonstrative examples– I chose to focus on figurative animations in the performance. I felt this quality would make it easier to communicate the performer's inner mental and emotional state by drawing elements that the audience could recognize and potentially relate to – e.g. a house, a bird, a face, a flower.

Both experts who reviewed the performance described interest in the capacity of Megafauna to enable the performer to tell a story and expectations that the performance itself more clearly conveys a narrative structure. They both commented positively on the elements of storytelling that emerged from embodied interactions – e.g. seeing a child being drawn on the performer's stomach and the performer's hand acting as a screen for the animation of a flower. However, both James and Miwa commented on the fact that the authoring interface was distracting them from the performance narrative. James felt seeing the Megafauna interface was particularly disruptive from a storytelling standpoint.

He described how seeing it appear and re-appear drew his focus away from the narrative of the piece. He also described how additional pauses in which the performer engaged in authoring–e.g moments of “silence”– would have improved his experience by enabling him to be transported by the performance story. Miwa would have wished to see more world-building of the visual scape. She described, for instance, how adding motion paths to the animated sequences could create additional movement to support better storytelling.

I designed Megafauna to support figurative imagery, however, I did not include features to explicitly support the creation of narrative performance. It also was not my intention to convey a narrative arc in the performance. Yet much of the feedback from James and Miwa was centered on the narrative potential of Megafauna, and a critique of the points of the performance that disrupted their immersion or understanding of the story elements. This feedback suggests that the use of figurative drawings in a live performance scenario can automatically create narrative expectations from the audience. Furthermore, these comments draw attention to the way transparency in the authoring process- in the form of visible interface authoring elements may actively conflict with the ability of the performer to convey a sense of immersion and preserve narrative flow.

These observations suggest that future research in procedural-manual systems would benefit from exploring how features could support narrative performance. One approach would be to develop mechanisms that could act on changing visual and sonic qualities – e.g. line style and colors or tones and timbers – or on organizing events in time to create movement and directionality established by a predetermined script as demonstrated by [56]. In Megafauna, features that act on the timelines show simple forms of how narrative can be supported by computational means. However, I also realize that crafting narratives requires advanced planning. I envision, for instance, a computational system for storytelling that could enable artists to create narrative elements– e.g. characters, settings, storylines– in advance and then improvise with them in a live performance context.

10.3 Real-time authoring audiovisual performance is suitable for embodied interactions

The performer's body is an essential expressive component in many non-computational forms of performance including dance, theater, and analog instrumental performance. Creating a performance with Megafauna provided me with the unexpected opportunity to include the performer's body as an additional modality to work with in computationally augmented forms of performance.

Interestingly, the expert reviewers found that the most successful elements of the performance were sections in which the projected visuals were overlaid on the performer's body, or the performer interacted physically with the visuals. Given that I did not originally develop Megafauna to support interactions between the performer's body and the projected animation, the positive response from the expert reviewers on these aspects led me to examine how the features of the Megafauna system supported these *embodied* interactions.

Some insight into this question can be drawn from my experience creating the performance itself. As I developed the performance, I found that the direct manipulation aspects of Megafauna made it convenient to specify the position and scale of projected animations relative to physical objects in projection space. For instance, the system provided, the flexibility to project on objects that changed position in space such as the performer's body, and offered the possibility to iterate quickly in order for the performer to determine the desired scale and position for each animation. Using a purpose-built projection mapping software like MadMapper [57] would have required additional manipulations each time the projection setting changes. Mapping software works by defining projection areas and associating video files to each of these areas. Each time the setting changes, the shape of these regions must be modified manually. In Megafauna, because the visual content is hand-drawn and can take arbitrary shapes, the mapping happens at the same time as the content gets created and therefore gets defined by the content automatically.

Building on these observations, I can envision how my approach in the Megafauna system might be further augmented to better support live artworks that integrate manual drawing with physical embodied performance. In the example works that I surveyed when defining the design space, I identified Miwa's performance (Fig 7.1a) and the Rhizomatics dance with drones performance (Fig 7.1c) to be the pieces presenting the highest level of embodied interactions. Both artworks are highly choreographed and require music to enable the synchronization of the human bodies with the visual components involved in these performances. By contrast, my experience of making a performance with Megafauna and the expert review feedback point toward live-mapping of hand-drawn animations as a possible space to develop in the field of audiovisual performances due to the speed afforded by making visuals through drawing and the immediacy of direct-manipulation systems. I also envision how my system could be used collaboratively, where a visual artist could draw on the body of other performers in real-time.

Chapter 11

Limitations

My system is limited to demonstrative applications, autobiographical use, and expert review. Demonstrative applications are a common approach to evaluation in HCI toolkit evaluation; they help reveal the limits of a system and the "paths of least resistance for authoring" [52]. The variety of my demonstrations suggests Megafauna has broad expressive potential. Usability evaluations with external artists would likely yield additional insights into the interpretability of Megafauna's primitives and the usability of the system interface.

I focus on self-use as another method of evaluation. I selected self-use in part because it enabled the creation of finished artwork over a multi-week creation period. Studies with external users frequently do not provide ample time for users to learn and familiarize themselves with tools to create expressive outcomes [58, 59]. Conducting a multi-week training and authoring session with experts to produce polished outcomes was beyond the scope of my current study. The integration of artistic production also builds on the unique opportunities of authorial artistic creation as a lens for HCI research [60].

My self-use approach shares many of the limitations of autobiographical design. Autobiographical methods are limited with respect to the subjectivity inherent to first-person research. Autobiographical design cannot always be generalizable to broader communities [49].

I rely on the evaluation of the output of my design activities by experts who are prominent in the field of audiovisual production as a means to externally judge how the functionality of Megafauna translates to compelling creative works.While I see opportunities for conducting longitudinal studies with expert practitioners to create polished work in future research, such a study was beyond the scope of this paper. Rather, I focus on the development and presentation of a novel tool, and the expert evaluation of a polished outcome created by the authors. In their survey on creativity support user evaluations, Remy et al. describe how user studies with human subjects may aid in understanding the usability of a system including error prevention and efficiency, but do not necessarily aid in assessing the degree to which a tool supports creative work [59]. My review is currently limited to two experts. Future assessments of the expressive potential of my system with a larger number of audiovisual artists are critical in understanding how Megafauna may support a broader range of animation and performance outputs.

Chapter 12

Conclusion and Future Work

Informed by interviews with professional animators, I developed Megafauna, a direct manipulation system inspired by live-coding frameworks that enables the creation and synchronization of hand-drawn visuals with audio. My evaluation showed that my system can be used to create improvisational and performative works. My system can also reproduce animation works that were previously made through analog and procedural means. In future work, I see opportunities to evaluate Megafauna with artists to further understand the process of learning the system and the expressive potential of my approach. I also wish to explore some development ideas highlighted by the expert reviews such as additional forms of automation in visual creation, structures that support the generation of more complex melodies, and the reduction of the visual presence of the user interface. Overall, I believe that continued research in new interaction methods for multimodal authoring will broaden expression and participation in both established and emerging artistic territories.

Bibliography

[1] G. Levin, *Painterly Interfaces for Audiovisual Performance.* PhD thesis, MIT, 1994.

[2] L.-B. Castel, *Ocular harpsichord*, 1734.

[3] T. Wilfred, *Clavilux*, 1919.

[4] B. Evans, *Foundations of a visual music*, *Computer Music Journal* **29** (2005), no. 4 11–24.

[5] M. McDonnell, *Visual music*, *Visual Music Marathon Program Catalog* (2007) 1–19.

[6] G. Wang and P. R. Cook, *2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument*, in *A NIME Reader* (A. R. Jensenius and M. J. Lyons, eds.), vol. 3, pp. 193–210. Springer International Publishing, Cham, 2017. Series Title: Current Research in Systematic Musicology.

[7] C. Roberts, M. Wright, J. Kuchera-Morin, and T. Höllerer, *Gibber: Abstractions for Creative Multimedia Programming*, in *Proceedings of the 22nd ACM international conference on Multimedia*, (Orlando Florida USA), pp. 67–76, ACM, Nov., 2014.

[8] A. McLean and G. Wiggins, *Tidal–pattern language for the live coding of music*, in *Proceedings of the 7th sound and music computing conference*, 2010.

[9] A. McLean, D. Griffiths, N. Collins, and G. Wiggins, *Visualisation of live code*, *Electronic Visualisation and the Arts (EVA 2010)* (2010) 26–30.

[10] B. Victor, *Dynamic pictures*, 2011. `http://worrydream.com/DynamicPicturesMotivation`.

[11] J. Jacobs, S. Gogia, R. Měch, and J. R. Brandt, *Supporting expressive procedural art creation through direct manipulation*, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, (New York, NY, USA), pp. 6330–6341, ACM, 2017.

[12] B. Shneiderman, *Direct manipulation: A step beyond programming languages (abstract only)*, in *Proceedings of the Joint Conference on Easier and More Productive Use of Computer Systems. (Part - II): Human Interface and the User Interface - Volume 1981*, CHI '81, (New York, NY, USA), p. 143, Association for Computing Machinery, 1981.

[13] B. Hempel and R. Chugh, *Semi-automated svg programming via direct manipulation*, in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), pp. 379–390, ACM, 2016.

[14] R. H. Kazi, F. Chevalier, T. Grossman, and G. Fitzmaurice, *Kitty: Sketching dynamic and interactive illustrations*, in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, (New York, NY, USA), ACM, 2014.

[15] E. Rosenbaum and J. Silver, *Singing fingers: fingerpainting with sound*, in *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC '10, p. 308–310, Association for Computing Machinery, Jun, 2010.

[16] S. Sen, K. Tahiroglu, and J. Lohmann, *Sounding brush: A tablet based musical instrument for drawing and mark making*, in *International Conference on New Interfaces for Musical Expression*, 2020.

[17] S. Bourgault and J. Jacobs, *Preserving hand-drawn qualities in audiovisual performance through sketch-based interaction*, in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 1–10, 2021.

[18] M. Harris, L. Husbands, and P. Taberham, *Experimental Animation: From Analogue to Digital*. Taylor & Francis, 2019.

[19] N. McLaren, "Pen point percussion." Canada: The National Film Board, 1951.

[20] T. Magnusson, *Herding cats: Observing live coding in the wild*, *Computer Music Journal* **38** (Mar, 2014) 8–16.

[21] A. McLean, *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*, p. 63–70, ACM Press, 2014.

[22] N. Collins, A. McLean, J. Rohrhuber, and A. Ward, *Live coding in laptop performance*, *Organised Sound* **8** (Dec., 2003) 321–330. Publisher: Cambridge University Press.

[23] G. Wang and P. R. Cook, *2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument*, vol. 3 of *Current Research in Systematic Musicology*, p. 193–210. Springer International Publishing, 2017.

[24] O. Jack, *Hydra.* May, 2021.

[25] J. McCartney, *Rethinking the computer music language: Supercollider*, *Computer Music Journal* **26** (Dec, 2002) 61–68.

[26] M. Puckette, *The patcher*, in *Proceedings of the 1988 International Computer Music Conference. San Francisco*, International Computer Music Association, 1988.

[27] Cycling74, *Max*, 2017. `https://cycling74.com/products/max/`.

[28] vvvv group, *vvvv*, 2017. `https://vvvv.org/`.

[29] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, *The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces*, in *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*, (Baton Rouge, Louisiana), p. 139, ACM Press, 2007.

[30] J. Patten, B. Recht, and H. Ishii, *Audiopad: a tag-based interface for musical performance*, in *Proceedings of the 2002 conference on New interfaces for musical expression*, pp. 1–6, 2002.

[31] B. A. Myers, *Taxonomies of visual programming and program visualization*, *Journal of Visual Languages & Computing* **1** (1990), no. 1 97–123.

[32] Z. Street, A. Albornoz, R. Bell, G. John, O. Jack, S. Knotts, A. Mclean, N. C. Smith, A. Tadokoro, J. S. van der Walt, and G. R. Velasco, *Facilitating technical connections and shared language between visualist and musician livecode performers, and developing practices that enable smoother collaboration*, in *Proceedings of the Fourth International Conference on Live Coding (ICLC)*, pp. 84–89, 2019.

[33] N. Collins and F. Olofsson, *Klipp av: Live algorithmic splicing and audiovisual event capture*, 2006.

[34] R. Bell and J. Chicau, *A trans-disciplinary tool for collaborative, choreographed, and embodied audio-visual live coding*, in *Proceedings of the Third International Conference on Live Interfaces (ICLC)*, pp. 173 – 180, 2018.

[35] I. E. Sutherland, *Sketchpad a man-machine graphical communication system*, *Transactions of the Society for Computer Simulation* **2** (1964), no. 5 R–3–R–20.

[36] A. McLean, *acid sketching*, 2015.

[37] A. Stead, A. Blackwell, and S. Aaron, *Graphic score grammars for end-users*, in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pp. 176–179, 2012.

[38] *SoundBow*. Binaura, 2016.

[39] M. Zadel and G. Scavone, *Different strokes: a prototype software system for laptop performance and improvisation*, in *Proceedings of the 2006 conference on New interfaces for musical expression*, pp. 168–171, 2006.

[40] N. N. Correia and A. Tanaka, *Avui: Designing a toolkit for audiovisual interfaces*, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, p. 1093–1104, ACM, May, 2017.

[41] B. Victor, *Stop drawing dead fish*, in *ACM SIGGRAPH 2012 Talks*, SIGGRAPH '12, 2012.

[42] B. Victor, "Inventing on principle." The 11th Canadian University Software Engineering Conference (CUSEC 2012) - Turing Complete, 2012.

[43] A. Blackwell, *Palimpsest: A layered language for exploratory image processing*, *Journal of Visual Languages and Computing* **25** (2014) 545–571.

[44] Audio Kit Pro, *Audiokit*, 2021.

[45] L. Muller and K. Kwastek, *The manual input workstation documentary collection: Interview with golan levin*, 2009. `https://www.fondation-langlois.org/html/e/page.php?NumPage=2221`.

[46] A. Pitaru and Z. Gage, "Sonic wire sculptor."

[47] *Orca*. Hundredrabbits, May, 2021.

[48] J. Zimmerman, J. Forlizzi, and S. Evenson, *Research through design as a method for interaction design research in hci*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), p. 493–502, Association for Computing Machinery, 2007.

[49] C. Neustaedter and P. Sengers, *Autobiographical design in hci research: designing and learning through use-it-yourself*, in *DIS '12*, 2012.

[50] K. Cascone, *Grain, Sequence, System: Three Levels of Reception in the Performance of Laptop Music*, *Contemporary Music Review* **22** (Dec., 2003) 101–104.

[51] S. Hsueh, S. F. Alaoui, and W. E. Mackay, *Understanding kinaesthetic creativity in dance*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, p. 1–12, ACM Press, 2019.

[52] D. Ledo, S. Houben, J. Vermeulen, N. Marquardt, L. Oehlberg, and S. Greenberg, *Evaluation strategies for hci toolkit research*, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), p. 1–17, Association for Computing Machinery, 2018.

[53] J. Jacobs, D. Mellis, A. Zoran, C. Torres, J. Brandt, and T. J. Tanenbaum, *Digital Craftsmanship: HCI Takes on Technology as an Expressive Medium*, in *Proceedings of the 2016 ACM Conference Companion Publication on Designing Interactive Systems*, (Brisbane QLD Australia), pp. 57–60, ACM, June, 2016.

[54] J. Li, S. Hashim, and J. Jacobs, *What we can learn from visual artists about software development*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[55] J. Jacobs, J. Brandt, R. Mech, and M. Resnick, *Extending Manual Drawing Practices with Artist-Centric Programming Tools*, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2018.

[56] H. Xia, *Crosspower: Bridging Graphics and Linguistics*, in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, (Virtual Event USA), pp. 722–734, ACM, Oct., 2020.

[57] *MadMapper*. MAD, 2022.

[58] D. R. Olsen, *Evaluating user interface systems research*, in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, (New York, NY, USA), p. 251–258, Association for Computing Machinery, 2007.

[59] C. Remy, L. MacDonald Vermeulen, J. Frich, M. M. Biskjaer, and P. Dalsgaard, *Evaluating creativity support tools in hci research*, in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, (New York, NY, USA), p. 457–476, Association for Computing Machinery, 2020.

[60] M. Sturdee, M. Lewis, A. Strohmayer, K. Spiel, N. Koulidou, S. F. Alaoui, and J. Urban Davis, *A plurality of practices: Artistic narratives in hci research*, in *Creativity and Cognition*, Camp;C '21, (New York, NY, USA), Association for Computing Machinery, 2021.