Haptic Experience Design

Tools, Techniques, and Process

by

Oliver Stirling Schneider

B.Sc. Honours, University of Saskatchewan, 2010

M.Sc., University of British Columbia, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

The University Of British Columbia

(Vancouver)

June 2016

Abstract

Haptic technology, which engages the sense of touch, offers promising benefits for a variety of interactions including low-attention displays, emotional connections, and augmented media experiences. Despite these advantages and an increasing presence of physical devices in commercial and research applications, there is still little support for the *design* of engaging haptic sensations. Previous literature has focused on the significant challenges of technological capabilities or physical realism, with limited development on supporting experience design.

In this dissertation, I ask the following question: **how can we design, build, and evaluate interactive software to support haptic experience design (HaXD)?** I have two goals: 1) *describe* HaXD, including processes, strategies, and challenges, to understand requirements; and 2) *prescribe* guidelines on designing, building, and evaluating interactive software that facillitates HaXD. To accomplish these goals, I will iteratively design three vibrotactile authoring tools, each a case study covering a different user population, vibrotactile device, and design challenge, and use them to observed HaXD with their target users. I then plan to make these in-depth findings more robust in two ways: generalizing results to a breadth of use cases with side-projects, and grounding them with expert haptic designers through interviews and a workshop. By capturing haptic experience design and creating guidelines for supportive tools, I hope to make a first step towards establishing haptic experience design as its own field, akin to graphic and sound design.

Revision: r0.1

Preface

No creative work occurs by a lone individual; this dissertation is no exception. All of the projects described in this work are collaborative efforts in at least some capacity. Even where the author contributed all work, there was often informal feedback from friends, family, and lab mates. As such, this dissertation will use the first-person plural, "we", throughout. In this preface, we clarify the author's contribution to the work, much of which has been published.

In Chapter 1, Chapter 2, and Chapter 9, Oliver contributed writing and framing, with feedback provided by the supervisor (Dr. Karon MacLean) and supervisory committee (Drs. Ronald Garcia and Michiel van de Panne) throughout his PhD program. Some of this thinking is combined with a handbook chapter currently under review, developed with Dr. Karon MacLean and Hasti Seifi: TODO.

In Chapter 3, Oliver contributed all work and ideas, with feedback and influence from supervisor Dr. Karon MacLean. This work has been published as .

in Chapter 4, Oliver contributed all work and ideas, other than initial interviews with designers and haptic experts. This work was conducted while on internship at Disney Research Pittsburgh, with some supplementary work done at UBC, with feedback and influence from internship supervisor Dr. Ali Israr and PhD supervisor Dr. Karon MacLean, and has been published as .

In Chapter 5, Oliver contributed all work and ideas, with feedback and influence from supervisor Dr. Karon MacLean, except for subsection TODO, where additional development work was done by TODO, TODO, and TODO. The majority of this work has been published as ; Section TODO is currently in development for a peer-reviewed submission.

In Chapter 6, Oliver was part of a collaborative team together with PhD student Hasti Seifi, undergraduate summer student Matthew Chun, and master's student Salma Kashani, all supervised by Dr. Karon MacLean. Oliver and Hasti planned and managed the project, with Matthew and Salma doing proxy design, study design, and data collection for low-fidelity proxies and visual proxies respectively. Oliver lead paper writing and quantitative analysis, working closely with the other authors, and presented the work published as .

In Chapter 7, Oliver played different roles depending on the focused design project.

Section TODO, FeelCraft Oliver worked closely with Siyan Zhao, supervised by Ali Israr. Oliver implemented the rendering system (which was co-developed with the engine described in Chapter 4), developed the MineCraft plugin and connection architecture, and wrote the AsiaHaptics paper with feedback from Ali Israr. Oliver and Siyan together designed the implemented feel effects (Oliver lead implementation), planned, shot, and edited the video submissions (Siyan lead editing); each presented the demo once (Oliver at AsiaHaptics 2014, Siyan at UIST 2014). Artistic contributions to the video were made by Kyna McIntosh and Madeleine Varner. This work has been published as .

Contents

List of Tables

List of Figures

Acknowledgments

Thank those people who helped you.

Don't forget your parents or loved ones.

You may wish to acknowledge your funding sources.

Chapter 1

Introduction

Human beings are physical, social creatures, and our technology has only just started to communicate on our terms. Over the years, computing has progressed from symbolic, machine-focused communication like punch cards, assembly languages, and terminal interfaces to physical and natural user interaction. Yet despite embracing new interaction techniques like touchscreens and voice control, the rich senses of touch have been relegated to buzzing alerts or limited to high-stakes expert systems like laparoscopic surgery.

Haptic technology involves the senses of touch; here we refer to both tactile (skin-based) and kinaesthetic (force- and position-based) perception. Between the resurgence of consumer virtual and augmented reality (VR & AR), rapid development of personal fabrication techniques, and recent additions of high-fidelity haptics to wearable products like the Apple Watch, we are poised to see haptic technology move from niche roles into mainstream adoption. This diverse field has been active in creating new devices and understanding human perception for decades, but the development of haptic media and design of haptic experiences remain a critical challenge. Haptic experiences are rich, diverse, multimodal entities which necessitate in-person interaction and have limited infrastructure. How can we enable creativity with these experiences? In this dissertation, I study the process of haptic experience design ("HaXD") and establish guidelines for building interactive software systems to support it.

1.1 Haptic Experience Design (HaXD)

We define "haptic experience design" (HaXD) as:

> *The design (planning, development, and evaluation) of user experiences intentionally involving both interactive technology and one or*

more perceived senses of touch, possibly as part of a multimodal or multisensory experience.

We use HaXD instead of the more general "haptic design", which can also refer to design practices related to haptics but not directly involving the user experience, e.g., mechanical design of a new actuator or software design of a new control method. Our definition also includes pseudo-haptics [**?**] and other illusions that trick the user into thinking haptic feedback is occurring without direct tactile or kinesthetic stimulation. For brevity, we will use "haptic designers" to refer to haptic *experience* designers.

Here, we also take a systems approach to design. Designers do not exist in a vacuum, but rather in in a physical, social, and cultural context, and are shaped by their personal experiences. As we will elaborate, diverse activities are involved in design, including *browsing* examples, *sketching*, *refining*, and *sharing*. Just as a user's physical, social, and cultural context must be considering in an interactive experience, so too must a designer's.

1.2 Approach

While many tools exist to support design in other modalities, such as graphic design, there are few for haptics. Part of this comes from immaturity of the field and lack of market penetration of highly expressive haptic devices. However, there are also intrinsic challenges to designing for the sense of touch. I approach this problem with three different strategies:

1. **Depth: Vibrotactile design tool case studies.** To understand design, I take a design perspective. In each of three case studies, I design, build, and evaluate a tool to support an aspect of haptic experience design, scoped to *vibrotactile* (VT) design. Each of these results in concrete implications for designing tools and a small window onto the larger HaXD process. Contributions include algorithms, data structures, interaction techniques, features, analytic techniques, and working software tools that have been employed by designers. Chapter 3, Chapter 4, and Chapter 5 outline these.

2. **Breadth: Focused haptic design projects.** While the case studies provide an in-depth investigation into vibrotactile sensation design, results may not generalize to other devices, and provide limited investigation into design activities like sharing. To generalize from vibrotactile effects, explore other aspects of haptic design, and gain personal experience as a haptic experience designer, I participate in several smaller side projects. These more focused

projects lend a broader context to our findings. Chapter 6 and Chapter 7 discuss these projects.

3. **Ground: Data from haptic experience designers.** Finally, despite the recent growth of the field, haptic designers remain relatively rare and difficult to recruit. To complement my primarily design-based approach and ground it with haptic experience designers in the field, I draw from other data sources: a workshop held at World Haptics 2015 and interviews with haptic designers. Described in Chapter 8, this synthesized contribution provides additional concreteness to the characterization of haptic experience design.

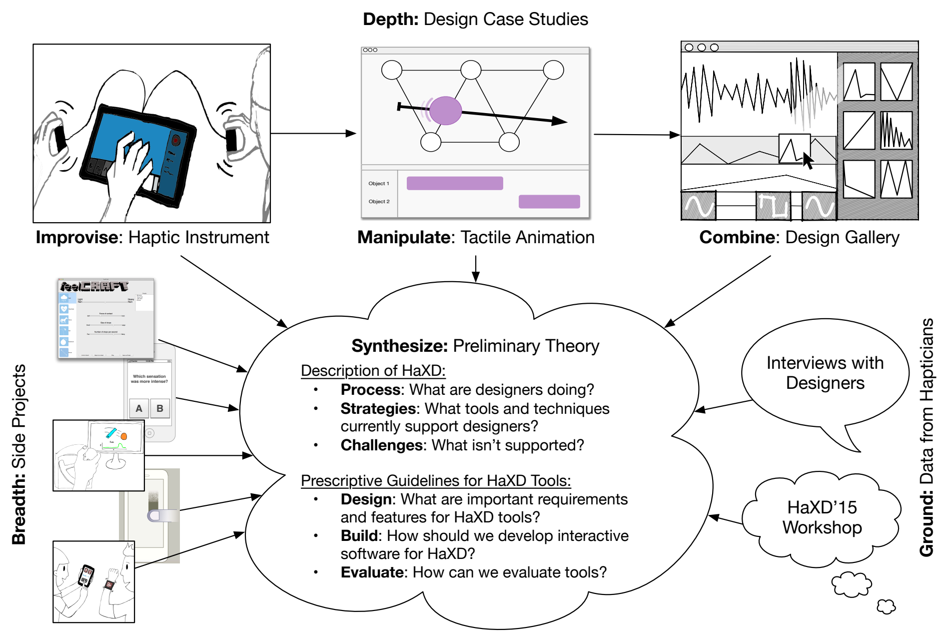


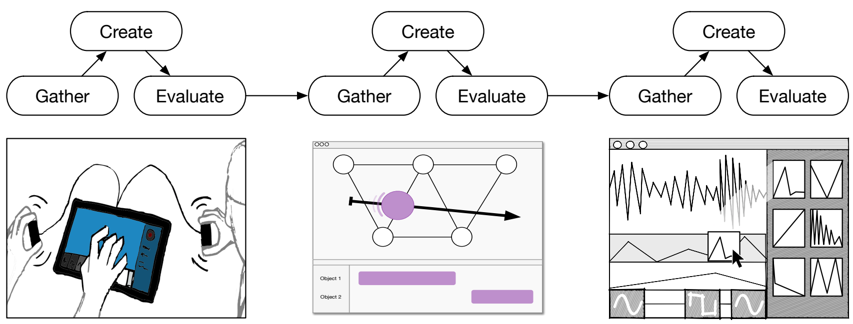
Figure 1.1: Approach overview. Three case studies investigate VT tools in-depth; findings are synthesized with side projects and grounded data into a preliminary theory.

1.2.1 Depth: Vibrotactile Design Tool Case Studies

Each case study investigates a different set of design concepts with varying user populations, VT device, and design challenges (Figure 1.2), but restricts scope to

VT sensations. This offers a deep look into an expressive and increasingly common class of haptic devices, allowing us to explore critical features in a somewhat controlled fashion. An iterative approach allows us to refine ideas and methods, and so each case study follows three steps: *gather*, finding requirements and previous design elements; *create*, where we design and build the tool; and *evaluate*, where we test the tool with its target population and consolidate lessons learned.

In Study 1, the Haptic Instrument, we focus on real-time, rapid design of VT sensations with a first look into themes of real-time design and collaboration. When participants worked with our tool, mHIVE (a "mobile Haptic Instrument for Vibrotactile Exploration"), compositions couldn't be edited, suggesting mHIVE was suitable for exploration and improvised communication, but not as suited to refining ideas. This informed Study 2, Tactile Animation, where we developed a single abstracted animation object directly manipulated in both space and time. Animators found our tactile animation tool, Mango, easy-to-use, and confirmed our findings about the value of real-time exploration.



	Improvise: Haptic Instrument	Manipulate: Tactile Animation	Combine: Design Gallery
Who	Haptic Designer-Proxies *(researchers with some haptics experience)*	Animators *(visual designers who would be haptic designers)*	End-users *(people with and without basic media editor experience)*
What	Dual Haptuators	Spatial Vibrotactile Pad	Phone-mounted Haptuator
How	Real-time, collaborative parameter control	Direct space/time manipulation of animation object	Example-based interaction techniques, morphing
Discovery	Good for ad-hoc exploration, collaboration; bad for refinement	Real-time manipulation valuable; animators drew from examples	Hypothesis: *Designers rely heavily on examples early on, make final adjustments directly*
Platform	Mobile (Android)	Desktop (Python)	Web (JavaScript)

Figure 1.2: Vibrotactile design case studies. Each studies an aspect of vibrotactile design with a varied set of users, devices, platforms, and foci.

One stand-out result from both Mango and mHIVE is that designers drew from their experience or examples found in the world, and wanted to re-use what they had created (e.g., through copy and paste). In Study 3, I explore the role of examples in haptic design with a web-based tool, "Macaron", a vibrotactile track-based editor with visible, incorporable examples directly embedded in the interface. We found examples were used primarily as templates to inform initial design, making each individual design easier but also scaffolding the user's understanding of how to create VT effects.

1.2.2 Breadth: Focused Haptic Design Projects

Each case study provides concrete knowledge for building a vibrotactile authoring tool, and some insight into the vibrotactile design process. However, haptic technology consists of many devices and experiences beyond vibrotactile.

FeelCraft and Feel Messenger are collaborations with Disney Research members Ali Israr and Siyan Zhao, looking at distributing and customizing haptic effects in a consumer setting with low-fidelity rumble motor devices. I take a haptic designer role to gain a personal understanding of the process, and a software engineer role to understand relevant architectures.

CyberHap is a collaboration between UBC and Stanford looking at force-feedback devices in education; a large team is involved with undergraduate Gordon Minaker leading development of a teaching interface since February 2015, co-supervised by PI Dr. Karon MacLean and me.

CuddleBit is a project inspired by the Haptic Creature and CuddleBot project. A small, breathing and vibrating robot will be designed along with a behaviour prototyping tool in summer 2015. I supervise undergraduate Paul Bucci in this project exploring multiple modalities and potential for receiving input through a sensor.

HapTurk is a collaboration with PhD candidate Hasti Seifi on different techniques to crowdsource feedback on VT icons. Master's student Salma Kashani and undergraduate Matthew Chun are developing visualizations and low-fidelity VT icons during summer 2015.

RoughSketch is a painting application for the TPad Phone, a variable-friction mobile device, for the World Haptics 2015 Student Innovation Challenge. Undergraduates Brenna Li, Paul Bucci, and Gordon Minaker are all fellow team members. Variable friction is a significant contrast to VT sensations as it is intrinsically connected to input: no sensation can be felt without active movement by the user.

1.2.3 Ground: Data from Haptic Experience Designers

I will synthesize findings from the three design case studies together with a number of side projects, the design literature, community feedback from a workshop on haptic experience design, and interviews with haptic designers into a preliminary design theory on how to support the creation of engaging, captivating haptic experiences. I expect to make progress on the following questions:

1. **Description of the Haptic Design Process.** What are the major **processes and tasks** conducted by haptic designers? What **strategies** do haptic designers employ, including existing tools? What are the **challenges** haptic designers face?

2. **Prescriptive Implications for HaXD Tools.** What are major **requirements** and **features** for designing HaXD tools? What are some considerations when **implementing** HaXD tools in software? How can we **evaluate** design tools effectively?

1.3 Outline and Contributions

This dissertation continues as follows. First, in Chapter 2, I cover related work with an overview of haptic technology and applications, a presentation of existing haptic design tools, and a discussion of design theory from other fields.

Then, I outline each VT design tool case study in Chapter 3, Chapter 4, and Chapter 5. In Chapter 3, we present findings from our first vibrotactile design tool, the haptic instrument, which supported easy exploration and informal feedback, but identified a key problem: lack of refinement for designs. In Chapter 4, we present findings from our second vibrotactile design tool, Mango, which established a generalized pipeline and was able to support both exploration and refinement for expert visual animators; it highlighted reuse as an important next step. In Chapter 5, we present findings from our third vibrotactile design tool, Macaron, which implemented a browsing interface and analytics system; we found examples played a large part of the design process, and that a web-based tool allowed for easy deployment.

I then describe focused haptic design projects in Chapter 6 and Chapter 7, and the results from our grounded data collection in Chapter 8. In Chapter 6, we document findings from HapTurk, a technique for getting feedback on vibrotactile designs at scale: from the crowd using proxy vibrations distributed over Mechanical

Turk; we also comment on uses for haptic broadcasting. In Chapter 7, we synthesize together findings from our side projects, showing generality by applying our understanding of haptic design explicitly in several domains and gaining practical experience designing haptic experience. In Chapter 8, we complement our design-based inquiry through interviews with professional haptic designers and a workshop run to elicit feedback from the community; this captures a description of haptic design, reinforcing our findings for important support tools, and identifies more systematic challenges.

Finally, in Chapter 9, we conclude with a summary of our final results and directions for future research.

Chapter 2

Background

In this chapter, I discuss haptic technology, haptic design tools, and non-haptic theory of design.

2.1 Haptic Technology

Haptic technology is typically separated into two classes based on the main sense modality: *tactile* sensations, perceived through the skin, and *proprioception*, or the sense of body location and force.

The most common consumer-facing tactile technology is vibrotactile (VT), where vibrations stimulate the skin. Eccentric mass motors (sometimes “rumble motors” or, when small, “pager motors”) are found in many mobile devices and game controllers. However, these have only a single degree of freedom: the input voltage or current, which corresponds to the combined amplitude and frequency of the device. Eccentric mass motors are cheap, salient, widespread, but inexpressive.

More expressive are tactors, or voice coils, implemented in a variety of ways. Behaving similar to small speakers, tactors offer two degrees of freedom: frequency and amplitude, and are typically more responsive than rumble motors. One of the more common and expressive is the C2 tactor, intended to directly stimulate the skin through contact or a thin membrane; the tactile animation project (**??**) uses an array of C2 tactors. Another common device, commonly used in research or prototyping, is the Haptuator [75]. Instead of directly stimulating the skin, this actuator typically shakes another device held by the user, such as a mobile device [76], pen [15], or other handle. Both the haptic instrument (Chapter 3) and Macaron editor (**??**) use Haptuators.

Proprioception, on the other hand, is synthesized from the muscle spindle and golgi-tendon organ (GTO), as well as tactile and visual cues [39]. Common devices include Geomagic Touch (previously the Sensable PHANTOM) and Falcon devices, offering three degrees-of-freedom: force in three directions and torque in three orientations. Simpler one degree-of-freedom devices are used in education, prototyping, and experimentation. Devices include the Twiddler (a haptic knob from UBC) and the HapKit (a low-cost paddle by the Stanford CHARM lab).

2.2 Haptic Design Tools

Currently, haptic designers have access to a limited number of hardware and software platforms, a set of philosophical design perspectives, and authoring tools.

2.2.1 Platforms

Many software libraries aim to support developers. The UPenn Texture Toolkit contains 100 texture models created from recorded data, rendered through VT actuators and impedance-type force feedback devices [15]. The HapticTouch Toolkit [41] and Feel Effect library [35] control sensations using semantic parameters, like "softness" or "heartbeat intensity" respectively. Vibrotactile libraries like Immersion's Haptic SDK (immersion.com) connect to mobile applications, augmenting Android's native vibration library. Force feedback devices have software platforms like CHAI3D (chai3d.org), H3D (h3dapi.org), and OpenHaptics (geomagic.com).

Hardware prototyping platforms like Arduino (arduino.cc) provide an open source microcontroller and development platform for physical prototyping. Phidgets (phidgets.com) facilitate rapid hardware prototyping with over 20 programming languages [24]. More recently, Wooden Haptics gives open-source access to fast laser cutting techniques for force feedback development [23], and faBrickation streamlines prototyping for 3D printing [53]. These platforms, especially Arduino, have made significant improvements to enable rapid iteration and hardware sketching. However, I believe we can do much better: these platforms require programming, hardware, and haptics expertise, and include inherent time costs like compilation, uploading, and debugging.

2.2.2 Design Perspectives

Some higher-level perspectives offer outcome targets or design attitudes to guide haptic practitioners. "DIY Haptics" categorize feedback styles and design principles [29, 46]. "Ambience" is proposed as one target for a haptic experience [47]. Haptic illusions can serve as concise ways to explore the sense of touch, explain concepts to novices and inspire interfaces [28]. "Simple Haptics", epitomized by *haptic sketching*, emphasizes rapid, hands-on exploration of a creative space [50, 51]. Haptic Cinematography [16] uses a film-making lens, discussing physical effects using cinematographic concepts. The notion of distributed cognition [32] has particular relevance for haptic design, suggesting that people situate their thinking both in their bodies and in the environment. Haptics courses are taught with a variety of foci including perception, control, and design, providing students with an initial repertoire of skills [38, 56].

Haptics has often made use of metaphors from other fields. Haptic icons [45], tactons [3], and haptic phonemes [21] are small, compositional, iconic representations of haptic ideas. Touch TV [49], tactile movies [40], haptic broadcasting [9], and Feel Effects [35] attempt to add haptics to existing media types, especially video.

Musical analogies have frequently been used to inspire haptic design tools, especially VT sensations. The Vibrotactile Score, a graphical editing tool representing vibration patterns as musical notes, is a major example [43, 44]. Other musical metaphors include the use of rhythm, often represented by musical notes and rests [5, 7, 10, 73]. Earcons and tactons are represented with musical notes [3, 4], complete with tactile analogues of crescendos and sforzandos [6]. The concept of a VT concert found relevant tactile analogues to musical pitch, rhythm, and timbre for artistic purposes [26]. Correspondingly, tactile dimensions have been also been used to describe musical ideas [18].

The language of tactile perception, especially affective (emotional) terms, is another way of framing haptic design. Many psychophysical studies have been conducted to determine the main tactile dimensions with both synthetic haptics and real-world materials [20, 55]. Language is a promising way of capturing user experience [54], and can reveal useful parameters, e.g., how pressure influences affect [77]. Tools for customization by end-users, rather than expert designers, are another way to understand perceptual dimensions [67, 68]. However, this work is far from complete; touch is difficult to describe, and some even question the existence of a tactile language [37].

2.2.3 Authoring Interfaces

As long as designers have considered haptic effects for entertainment media, they have needed compositional tools [26]. A great deal of previous work has focused on how to prototype or author haptic phenomena using non-programming methods.

Many user-friendly interfaces help designers create haptic sensations, especially with vibrotactile devices. The Hapticon editor [20], Haptic Icon Prototyper [71], and posVibEditor [62] use graphical mathematical representations to edit either waveforms or profiles of dynamic parameters (torque, frequency) over time. The Vibrotactile Score [44] was shown to be generally preferable to programming in C and XML, but required familiarity with musical notation [43]. The Demonstration-Based Editor [31] allows control of frequency and intensity by moving graphical objects on a touchscreen. Similar to the SPIN lab's Haptic Instrument (mHIVE, Chapter 3), this mobile tool was shown to be intuitive and easy to use for exploration or communication, but faltered when refining more elaborate sensations.

Commercially, Apple's end-user vibration editor has been present in iOS since 2011 (iOS 5) but only produces binary on/off timing information. Immersion provides two tools: TouchSense Engage is a software solution for developers, while Touch Effects Studio lets users enhance a video from a library of tactile icons supplied on a mobile platform. Vivitouch Studio allows for haptic prototyping of different effects alongside video (screen captures from video games) and audio, and supports features like A/B testing [72].

The control of multi-actuator outputs has been explored by TactiPEd [57], Cuartielles' proposed editor [14], and the tactile movie editor [40]; the latter combined spatial and temporal control using a tactile video metaphor for dense, regular arrays of tactile pixels ("taxels"), including a feature of sketching a path on video frames. However, these approaches embrace the separate control of different actuators, rather than a single perceived sensation produced by the multi-actuator device, which we address with tactile illusions in **??**.

2.3 Non-Haptic Design Theory

In this section, we present related work on non-haptic design organized into three major elements: problem preparation, hands-on design, and collaboration.

2.3.1 Problem Preparation

Creative tasks, like design, are often defined as the recombination of existing ideas, with a twist of novelty or spark of innovation by the individual creator [74]. Also known as the "problem setting" [66], "analysis of problem" [74], or "collect" [70] step, problem preparation involves getting a handle on the problem, drawing inspiration from previous work. Schön demonstrated that designers initially frame their problems before developing a solution [66]. Schön also describes the designer's repertoire, their collected experience, which aids in design. External examples are especially useful for inspiration and aiding initial design [8, 30], which we explore in **??**.

2.3.2 Hands-On Design

There has recently been a shift in how we interpret the act of thinking. No longer is thinking relegated to the head; cognition is now seen as being situated in the physical world [32]. The designer must iteratively generate a varied set of initial ideas (ideation) and then prune them (evaluation), repeating this step many times to settle on a single design [8]. Working with multiple ideas simultaneously is a boon to good design. Developing interfaces in parallel can facilitate generation and evaluation, delaying commitment to a single design [27, 59], while in groups, sharing multiple designs improves variety and quality of designs [17].

Sketching supports ideation, evaluation, and multiple ideas, allowing the designer to explicitly make moves in a game-like conversation with the problem [66]. It is so important that some researchers declare it to be the fundamental language of design, like mathematics is the language for scientific thinking [13]. The power of sketching, according to Cross, is contained in its ability to describe a partial model of a proposed design or problem. Detail can be subordinated, allowing a designer to zoom-in, solve a problem, and then abstract it away when returning to a high-level view. This has implications for software tools: designers must easily navigate the design space with undo, copy and paste, and a history of progress, creating tools with a "high ceiling" and "wide walls" [59].

2.3.3 Collaboration

Design is a collaborative process with the potential for generating more varied ideas [74], and is important for creativity support tools [59, 70]. Although sometimes group dynamics influence the design process negatively, proper group management and sharing of multiple ideas results in more creativity and better designs [30]. Shneiderman in particular has championed collaboration in design [70], and suggests two different types of collaboration to be supported by creativity tools: *relating*, informal discussions with colleagues, and *donating*, disseminating information to the public/annals of time. Orthogonal to these intended purposes (relating and donating) is the collaboration context. Computer-supported collaborative work often separates interactions into four contexts ordered into two dimensions: collocated (same location) or distributed (different locations), and synchronous (simultaneous) or asynchronous (at different times) [19]. Collaboration is notable because it is inherently challenging to haptic design: two people can look at the same image or hear the same sound from across a room, but touch is a local sense, far easier in a collocated, synchronous setting. We explore collaboration with the Haptic Instrument (Chapter 3) and the FeelCraft and HapTurk side-projects (described in **??**).

2.4 Methodology

⌈**OS** *FROM HAPTIC INSTRUMENT:*⌋ We collected and analyzed our data using the methodology of phenomenology, an established variant of qualitative inquiry used in psychology to investigate topics ranging from visual illusions to tactile experience [12, 54, 60]. Phenomenology explores subjective experience, appropriate for an investigation into the more intangible qualities of pleasantness and affect. At this point, the rich, inductive data of qualitative analysis is more valuable than a controlled experiment with statistical analysis.

In particular, we use the Stevick-Colaizzi-Keen method as described by Moustakas [52]. In-depth interviews are conducted with a small number of participants. The interviewer, Researcher 1 (R1), also documents his experience, as if he was interviewing himself. Then, R1 transcribes each interview, including his own. Transcripts are divided into non-overlapping, non-redundant statements about the phenomena known as Meaning Units (MUs). This considers every statement that the participants make, and does not discount any due to bias or selective searching. Then, MUs are clustered into emergent themes. ⌈**OS** *Describe methodology in more detail here.*⌋

Chapter 3

Improvise: The Haptic Instrument

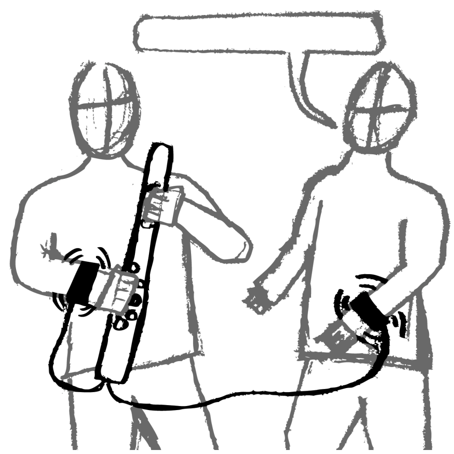


Figure 3.1: Concept sketch of a haptic instrument. Both users experience the same sensation, controlled in real-time.

The Haptic Instrument case study[1] is a first exploration into building a vibrotactile design tool, looking at design activities of exploration and informal sharing. Through it, we investigate the role of real-time feedback and synchronous collaboration on haptic experience design, using participants with some haptics experience, serving as proxies for haptic experience designers. Conventional haptic design tools contain a slow iteration, requiring programming or rendering before

[1]Published in Haptics Symposium 2014 [64] and at a CHI 2014 workshop [65].

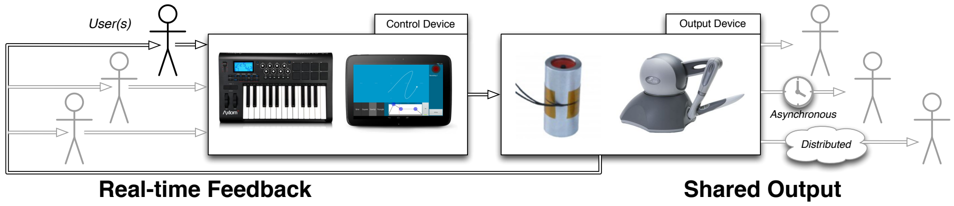


Figure 3.2: The haptic instrument concept. One or more people control the instrument, and receive real-time feedback from the device. Any number of audience members can feel the output in real time as well. Control methods can vary, from traditional musical control devices (such as the M-Audio Axiom 25, used in preliminary prototypes) to touchscreen tablets (used in mHIVE). Output devices vary as well.

playback. Using a music composition metaphor (as in [44]), we are writing music without ever playing a note: composing a work in its entirety, then listening to the result before making changes. In contrast, musicians often use their instruments as a tool for serendipitous exploration when designing music. Furthermore, music is collaborative, with communication facilitated by a reference point of a sound.

Our approach is to directly use a *haptic instrument*, inspired by musical instruments but producing (for example) vibrotactile (VT) sensations rather than sound (Figure 3.1). Haptic instruments are intended to have two main uses: they provide real-time feedback to the user to facilitate improvisation and exploration, and produce haptic output to multiple users as a *what-you-feel-is-what-I-feel* (WYFIWIF) interface. This allows for a dialogue that includes a haptic modality: haptic instruments create a shared experience of touch, allowing for a common reference point.

3.1 Design

There are several main design dimensions that can be considered in a haptic instrument (outlined in Figure 3.2). A haptic instrument can occupy multiple positions on these dimensions.

Asychronous/synchronous. Though a haptic instrument must provide real-time feedback, its collaborative (shared-output) aspect could be either synchronous (by having multiple people experience the real-time output) or asynchronous (by allowing for recording and playback, important for design).

Collocated/distributed. A haptic instrument's output could be present only for users in the same room, or be broadcast over a network to people around the world. For example, multiple mobile devices could all display identical output in a distributed manner.

Private/shared control. A haptic instrument's control could be private (operated by a one person at a time) or shared (multiple users control the display). Shared control could be collocated or distributed (*e.g.*, a web interface and shared object model).

Output mechanism. Each haptic instrument will control a haptic device, which has its own mechanism for providing a haptic sensation (*e.g.*, vibrotactile sensations). Because haptic devices can be complex and combine multiple mechanisms, this is a large space in its own right. Characterizing the different display mechanisms is something that we must leave to future work. Suffice it to say, a haptic instrument will be different depending on its output device.

Number of haptic instruments or output devices. One consideration is whether a haptic instrument is intended to operate alone, or with other haptic or multimodal instruments. One can imagine haptic jam sessions for inspiration and ideation, or even form haptic bands for artistic expression. This is highly related to private/shared control – there is a fine line between several identical haptic instruments with private control, and a single haptic instrument with shared control and several output devices. Note that a haptic instrument may involve several devices to produce shared-output.

Control mechanism. Similarly, a haptic instrument could be controlled in a variety of ways. From musically-inspired MIDI controllers to smartphone applications, we envision a wide variety of control methods. Even a real-time programming environment might be appropriate for complex interactive sensations, so long as the control mechanism works with the output device's paradigm.

3.1.1 mHIVE

We developed mHIVE, a mobile Haptic Instrument for Vibrotactile Exploration, to begin to explore how a haptic instrument should work and what it should do (Figure 3.3). mHIVE is a collocated, synchronous haptic instrument for a single user. It accommodates shared display via dual Haptuators [75] and is operated with a single-touch tablet-based interface for direct manual control (Figure 3.4). mHIVE is designed for for VT sensations, which are common, do not require interactive programming, are controlled through waveforms (analogous to music), and their low-level control parameters are well understood.

mHIVE offers real-time control of frequency, amplitude, waveform, envelope, duration, and rhythm, identified as the most important parameters for VT sensations [3, 6, 7, 26, 61]. mHIVE is implemented in Java using the Android SDK [2], and the FMOD sound synthesis library [22] to produce sounds, sent to two or more Haptuators through an audio jack. We deployed mHIVE on an Android Nexus 10

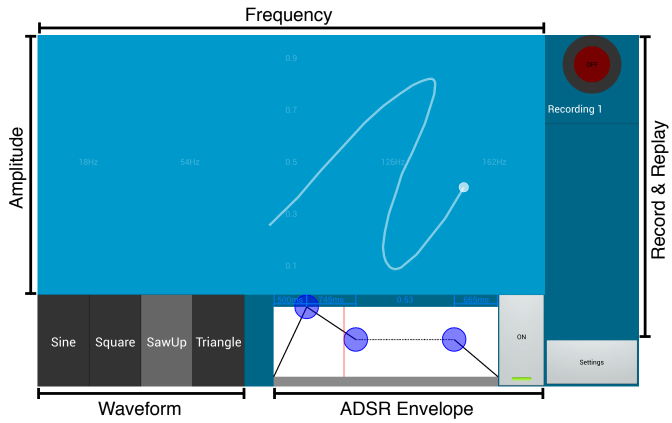


Figure 3.3: mHIVE interface. Primary interaction is through the amplitude-frequency view, where visual feedback is provided through a circle (current finger position) and a trail (interaction history).

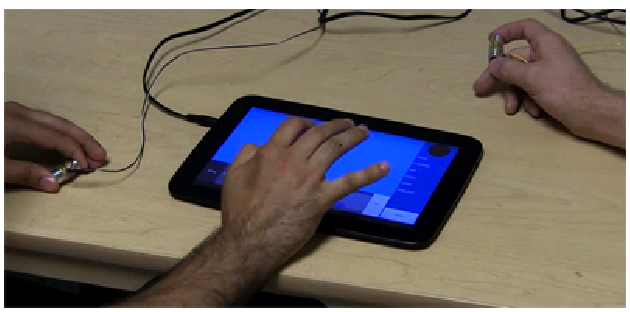


Figure 3.4: Study setup. Both the participant (left) and the interviewer (right) feel the same sensation as the participant controls mHIVE.

tablet running Android 4.2.1.

3.2 Study

We conducted a qualitative study to investigate two questions. First, is mHIVE an effective tool for the expression, exploration, and communication of affective phenomena? Second, what language, mental models, and metaphors do people use to describe vibrotactile sensations, and how do they relate to mHIVE's low-level control parameters?

One researcher collected and analyzed data using the Stevick-Colaizzi-Keen method as described by Moustakas [52]. Our 1-hour open-ended interviews used the following protocol:

1. Ask the participant for their background: occupation, experience with touch-screens, haptics, music, and video games.

2. Demonstrate mHIVE to the user, and invite them to explore while thinking aloud to describe the sensations they feel.

3. Probe the design space by asking participants to explore different control parameters, and to explore their metaphors (*e.g.*, if the participant describes a sensation as "smooth", R1 would ask them to try to produce a "rough" sensation).

4. Ask the participants to produce sensations for the six basic cross-cultural emotions documented by Ekman [**?**], and rank how well they think their sensation represents the emotion on a 4-point semantic differential scale (Very Poorly, Somewhat Poorly, Somewhat Well, Well). This was done both as an elicitation device to gather a wider range of interactions with mHIVE, and to directly investigate a design task.

5. Set the Haptuators down, and ask the participants to describe their experience of working with mHIVE in as complete detail as possible to evaluate the device itself.

R1 conducted the interviews and analysis, which required specialized knowledge of mHIVE. Scores of inter-rater reliability common with other qualitative analyses (*e.g.*, grounded theory [11]) are inappropriate and unavailable, as we did not conduct deductive, low-level coding. To improve reliability, R1's documented experience was analyzed first, and then consulted during analysis to remove bias (*e.g.*, to not use terms only used by the experimenter).

3.3 Results

We sought participants with experience designing haptics as a proxy for expert designers for our initial study. Four participants were recruited through email lists and word-of-mouth (P1-4, three male), and were all in the age range of 26-35 with self-reported occupations including graduate students or post-docs in information visualization, HCI, and human-robot interaction). All had experience working with haptic technology, and (because of this requirement) all knew the main researcher in a professional capacity, although only P2 had seen earlier prototypes of the haptic instrument. The small sample size, typical for phenomenological studies

[12], was appropriate for the rich data we wanted. Data collection ended when we achieved saturation of new results, and had a clear direction for our next iteration.

Here we report the three major themes that emerged during analysis: mHIVE's success as a haptic instrument, mHIVE's limitations that reveal more detail about the haptic design process, and the use of language in the study.

3.3.1 mHIVE Succeeds as a Haptic Instrument

Our results suggest that mHIVE can be effective for exploration of a design space, and communication in the haptic domain. Overall, mHIVE was well received, seen as a novel and promising tool. *"I definitely liked it"* (P1), *"I think there should be more devices like this for designing haptic icons"* (P2).

Serendipitous exploration. Participants reported that mHIVE was best served to explore the design space, generate a number of ideas, and try things out. Serendipitous discoveries and exclamations of surprise were common. Participants were able to *"accidentally stumble upon something"* (P2) as they explored the device. *"I felt I could get a large variety"*, *"I could easily play around with the high-level to find out what was neat"* (P3).

Communication. mHIVE established an additional modality for dialogue. The dual outputs created a shared context, demonstrated by deictic phrases: the additional context of the vibrotactile sensation was required to make sense of the statement. The use of "that" and "there", reminiscent of the classic "Put That There" multimodal interaction demo [**?**] indicate a shared reference point was established from the haptic instrument. *"So there'd be like, (creates a sensation on the device), which is pretty mellow"* (P3).

In particular, P4 successfully communicated the sensation of sleepiness to the R1, by asking whether R1 could guess the sensation. *"Can you guess it?"* (P4) *"Sleepy?"* (R1) *"Yeah. Pretty good"* (P4). The dialogue worked as a two way channel, as R1 was able to phrase questions using the device. *"It was different"* (P2) *"How was it different?"* (R1) *"You delayed the first part, it felt new"* (P2).

Certain sensations, like a feeling of randomness, could only be felt when another person controlled mHIVE. *"When someone else does it, I feel better, it's like, you cannot tickle yourself"* (P2).

3.3.2 Tweaking through Visualization and Modification

During analysis, some key directions for future design emerged around visualization and control capabilities.

Inability to tweak. Though mHIVE supported exploration and collaboration, we found it was inadequate as a standalone design tool. Few created sensations were considered to be final. Many descriptions were hedged and in the design task, few sensations captured the emotional content well. *"I dunno, maybe that's afraid?"* (P1), *"Still felt that you can make them better"* (P2), *"To me that's more fuming (laughing) than it is angry"* (P3). On some occasions, participants were certain about their descriptions. *"Sad, definitely down on the amplitude with sad... oh that's totally sad. Yeah."* (P1). This was uncommon, and usually tied to discovering an ideal sensation during the design task.

More visualization and recording. Part of mHIVE's inability to support tweaking was due to cognitive limitations for both memory and attention. Participants found it difficult to remember what they had tried before, and to pay attention to the output while simultaneously controlling it. *"There's a lot of variables which, when I'm trying to compare between two configurations... it was hard sometimes to remember what I had tried"* (P3), *"I definitely liked being able to feel a stimulus without having to implement it, you know, it allows me to focus more on what it feels like"* (P1).

Participants suggested that although visualization and recording features helped somewhat to overcome these limitations, more was needed. All requested greater emphasis on recording through repetition or looping, both to aid memory and allow for focus on the sensation independent of device control.

Allowing persistent, modifiable sensations and alternative visualizations could also help participants overcome these limitations. *"The recording records what I do, but it'd be nice to have it repeat stuff"* (P3), *"It might conceivably be nice to be able to, you know, draw a curve, draw a pattern, draw like you would in paint, and then be able to manipulate it, replay it, move the points, see what happens"* (P1).

3.3.3 A Difficult Language

Our study was too small to analyze language patterns in detail, but exposes emerging trends.

Pleasantness, ADSR, and frequency. Participants often started with a statement of like or dislike rather than a description. Pleasant sensations often involved the ramp-in and ramp-out ("echo" or "ringing") of the ADSR envelope, or lower-frequency sensations. Longer, higher frequency without ramp-in and ramp-out were less pleasant. *"I don't know how else to describe it, I kinda like it"* (P1), *"Yeah, this [ADSR] seems natural, somehow"*, *"It feels unnatural to kill the echo right away"* (P2), *"I like this [low-frequency] sensation cuz to me it feels a lot like purring"* (P3).

Waveform. Participants all noticed differences between waveforms, but were often challenged in expressing them (P4 used the musical term "timbre"). Square waves in particular were distinct, with a greater range and stronger affinity to mechanical sensations. *"It's interesting, they feel more different than I thought they would"* (P1), *"If you want to make something feel like a motorcycle, you would definitely need square wave"* (P2).

Aural/haptic metaphors drawn from previous experience. For the most part, participants used concrete examples and direct analogies to describe sensations, often drawn from their previous experiences. One stand-out strategy employed by all participants was onomatopoeias: *"beeooo"* (P1&4), *"vroom"* (P1), *"bsheeeooo"*, *"boom"*, *"neeeaa"*, *"mmmMMMmmm"* (P2), *"pa pa pa pa"*, *"tum tum tum tum"*, *"tumba tumba tumba tumba"* (P3); *"upward arpeggio, like, (singing with hand gestures) na na na naaa"* (P4). Other sound-based metaphors were very common, including hum, buzz, whistle, rumble (P1); bell (P1, P2); squeaky, creak (P2); or thumpy (P3). Still other descriptors were directly haptic in nature: rough, flat (P1); sharp, round, ticklish (P2); sharp, smooth, cat pawing (P3); impatient foot tapping (P4).

3.4 Discussion

⌈**OS** *TODO: modify and link this discussion to other chapters, once they are better established.*⌋ Here we interpret these themes to draw implications for haptic design tools, and compare to research on the language of haptics. We then reflect upon our methodology and limitations.

3.4.1 Design Tools

mHIVE was able to achieve the two main goals of a haptic instrument, facilitating both exploration and collaboration. Participants were clearly able to explore the different low-level parameters, and encountered serendipitous or unexpected sensations through improvisation. mHIVE created a shared experience that facilitated communication between R1 and the participants. We can thus conclude that haptic instruments are a promising new tool in a haptic designer's arsenal, with a first, successful implementation in mHIVE.

However, the second theme shows that serendipity and communication are only part of the equation. mHIVE does not serve as a general editor of haptic sensations. In particular, participants found their attention split when controlling the device and feeling the sensation; perhaps the real-time control should allow for a rapid, but not instantaneous, switch in focus between control and perception. More generally, participants were unable to tweak sensations because there was insufficient support for comparing ideas or evolving an existing idea.

In hindsight, this general difficulty is understandable given the broader context of the musical instrument analogy we used for inspiration. Musical instruments are not used to write songs on their own, but combined with notation or recording media. A similar combination of a haptic instrument and recording might be described more succinctly as a *haptic sketchpad*. Sketching is critical in design because it allows for the evolution of an idea through multiple sketches, as well as criticisms, comparisons, and modifications [**?**]. Emphasizing a history feature that supports multiple versions of sketches, the user could develop an idea as if with a multiple pages in a sketchbook. Haptic sketching in hardware has already been shown to be effective [51]. As well, a visual metaphor resonates with the desire for more effective visualization.

Ultimately, haptic instruments may be most useful as one element in a suite, or component of a more general tool. A haptic instrument could complement a graphical editing tool that does support tweaking, such as the vibrotactile score [43, 44] or the hapticon editor [20]. As part of a more comprehensive tool, mHIVE could be improved to reduce cognitive barriers to memory and attention. Alternatively, we could add functionality to mHIVE to support looping, visualization, and direct manipulation of the sensations within the tool. We will explore these options as we iterate on mHIVE's design in future work.

3.4.2 Language

Our preliminary results for language are compatible with the literature, supporting previous work. Participants' readiness to say whether a sensation was pleasant or not supports the view that touch is affective in nature, and that knowing what one likes or doesn't like is a primary function of touch [37]. ADSR pleasantness and high-frequency unpleasantness are both consistent with the literature: Zheng and Morell note that ramped signals influenced affect more positively than step signals, and 3s high-frequency sensations were annoying or agitating [77]. The heavy use of onomatopoeias is reminiscent of Watanabe *et al.*'s work with static materials [**?**]. However, in our study, onomatopoeias were often used to express dynamic sensations (beeeooo being a gradual decrease in amplitude and frequency), which might be a useful direction for future work.

3.4.3 Methodology and Limitations

Although phenomenology is uncommon in the haptics community (excluding [54]), we found it to be an effective way to empirically examine the subjective experience of using mHIVE. Because the community is still developing processes and tasks for haptic design, qualitative studies seem to be an especially appropriate way to tackle these problems. Once we have further defined haptic design, we can then move to more task-based, experimental methods.

Our study was a first round of feedback to inform our next iteration, and has limitations. First, our participant pool is (intentionally) small, and participants were all collected through our professional network, as people with haptic design experience are rare. As we continue to tackle the problem of haptic design, we hope to seek out a larger and more diverse pool of participants, and explore more realistic design tasks.

3.5 Discussion

Ultimately, mHIVE was able to achieve the two main goals of a haptic instrument, facilitating both exploration and collaboration, which showed value in real-time exploration and a shared output context. mHIVE also had limitations - participants could not edit sensations and found it difficult to keep track of multiple sensations. This is understandable given the broader context of the musical instrument analogy we used for inspiration. Musical instruments are not used to write songs on their own, but combined with notation or recording media. There may be no silver bullet with haptic design tools, with haptic instruments solving a particular set of processes (quick, easy ideation and communication for experts) but not others (final touches, distribution). Ultimately, haptic instruments may be most useful as one element in a suite, or component of a more general tool.

We follow-up on these leads in our subsequent design studies. In Chapter 4, we use a persistent model of a VT sensation for an editor, and confirm the value of real-time feedback while expanding the design palette to include spatial haptics. In Chapter 5, we attempt to mitigate the difficulty of describing haptics and draw upon examples by using a VT design gallery.

Chapter 4

Manipulate: Tactile Animation

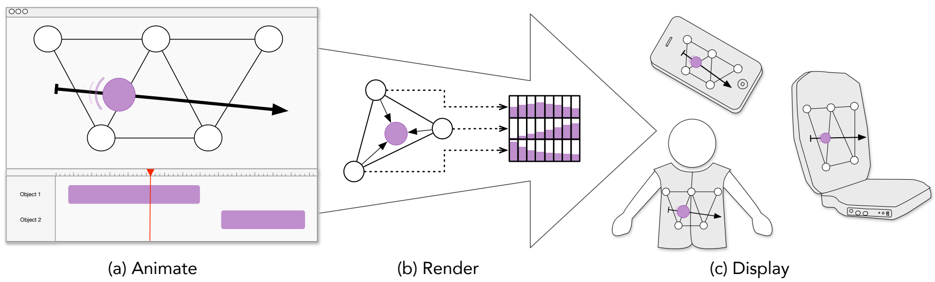


Figure 4.1: Concept sketch for tactile animation. An artist draws an animated sequence in the user interface and the user experiences phantom 2D sensations in-between discrete actuator grids.

In this second case study, we iterate on some of our findings from the haptic instrument to design a full authoring tool, using real-time feedback but supporting refinement with time-based editing. This work[1] targeted professional media designers (especially animators) creating spatial vibrotactile sensations. To afford real-time manipulation, we developed the *tactile animation object*, a persistent, manipulable primitive rendered through phantom vibrotactile sensations, and implement Mango, an editing tool built for animators. In our evaluation, professional animators found it easy to create a variety of vibrotactile patterns, with both experts and novices preferring the tactile animation object over controlling actuators individually. Furthermore, the tactile animation metaphor is a generalizable concept that can extend to several devices.

[1]In peer review at the time of this writing.

LR	Description
LR1	**Real-Time Playback** [51, 64] Rapid prototyping is essential for working with VT sensations, especially in absence of objective metrics. Feeling a sensation at design time allows iteration to converge faster to better results. However, *too* real-time can cause split attention.
LR2	**Load, save, manipulate** [59, 64**?**] A persistent object model is essential for sensation editing over longer projects and sharing with other designers or across devices. Well-defined actions upon a data structure also facilitates features like *undo* that support experimentation.
LR3	**Library of effects** [20, 30, 57, 71, 72] A library of saved sensations is an important feature used in previous haptic authoring tools, providing inspiration and preventing designers from re-inventing the wheel.
LR4	**Device configuration** [40, 43, 57**?**] Because of the many types of haptic devices, a general tool must be able to understand different devices. Lightweight configuration files are common in the literature, allowing users to select specific hardware, specify location and type of actuators, and choose a rendering algorithm.
LR5	**Multiple channels & combination of effects** [20, 57, 62, 71, 72] Being able to display multiple effects simultaneously, or combine effects via superposition or concatenation, is essential for expanding the design space. This is typically represented in a timeline, which represents the temporal behaviour of any objects.
LR6	**Visual/direct control metaphor** [14, 40, 57] Most previous tools consider each actuator separately. When thinking semantically about a spatial system, a direct view of the device and actuator layout is critical for direct manipulation.
LR7	**Audio/visual context** [40, 51, 72] Haptic perception depends greatly on additional senses [28]. By providing audio and visual feedback, these effects can be mitigated and the designer can experience haptic sensations in context.
LR8	**User Feedback** [64, 72] Receiving feedback from users, either by demonstration or A/B testing, is extremely valuable.

Table 4.1: Literature Requirements (LRs) for a tactile animation authoring tool.

4.1 Tactile Animation Authoring Tool

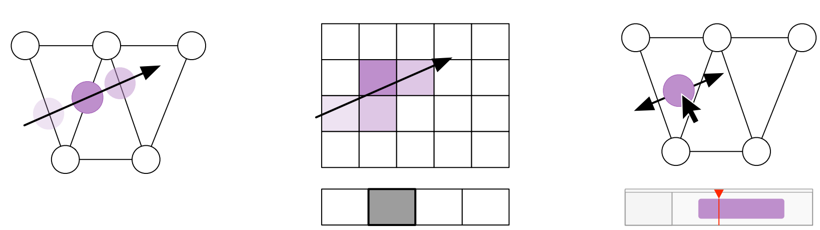
Our objective is to provide media designers with a familiar and efficient framework for creating dynamic haptic content. Mango's design is based on two sets of requirements: Literature ("LRs", Table 4.1), from prior research on haptic authoring tools, and Industry ("IRs") from interviews with five industry experts in haptic media creation and animation, which confirm and expand upon design decisions for other VT tools.

4.1.1 Gathering Design Requirements

We interviewed two industry experts with haptics experience from a media company (E1-2). E1 uses Max/MSP, OpenFrameworks, Processing, and Visual Studio to create haptic media. E2 is a professional media designer and an expert user of Pro Tools (an industry standard for authoring sound media). Together, E1 and E2 previously undertook a six-month training that included generation of dynamic haptic experiences on seats and supporting platforms using audio and video tools. Our interviews included meetings, recordings, and sketches of their experience during training.

In addition, we conducted contextual interviews of three industry animators (A1-3) interacting with non-tactile animation tools using a think-aloud protocol. A1 and A3 used Adobe After Effects, while A2 used Maya. A1 and A2 were tasked with creating an animation of two balls moving; A3 created an animation based on a sound file. These interviews yielded rich detail that we compiled into categories, then compared with our LRs (Table 4.1). LRs 2-7 also emerged independently from this stage. We extend the LRs with additional expert-drawn **industry requirements (IRs):**

IR1 - Animation window allows users to draw tactile animation objects, control them in space, and define their motion paths. The window is overlaid with location



(a) Tactile Brush [34]: precomputed paths

(b) Tactile Video [40]: frames of tactile pixels

(c) Tactile Animation: direct manipulation

Figure 4.2: Comparison between related systems.

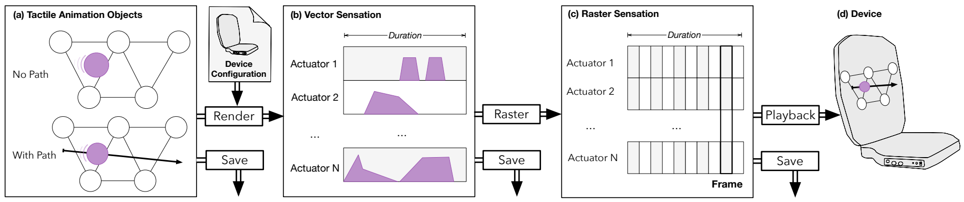


Figure 4.3: Tactile animation rendering pipeline. Users can: (a) create tactile animation objects; (b) render objects to actuator parameter profiles (such as amplitude) with our rendering algorithm; (c) rasterize vector sensations into frames; (d) play the sensation on the device.

and type of haptic actuators, providing visual feedback (LR8).

IR2 - Timeline is a time track for a tactile animation object. During playback, the animation is played on IR1 showing the movement of the animation relative to the tactile object. Object behaviours are linked to time track to visualize temporal variations. Time tracks are editable by inserting key frames.

IR3 - Object tools extend LR2, supporting direct manipulation operations on tactile objects such as “new”, “scale”, “translate”, analogous to object creation and manipulation in After Effects and Maya.

IR4 - Path tools define motion paths of tactile objects (straight lines, curves, input-device traces), and store them in a path library (LR3).

IR5 - Haptic rendering schemes compute output waveforms for each actuator channel, animated visually in the animation window. Users select the scheme from a list for connected hardware, defined in a hardware configuration file (LR4).

IR6 - Global parameter tools allow the user to control the overall feel of the tactile animation object. Analogous to filters and effects applied on the object, this includes parameter setting for frequency, intensity and modulation.

We developed a tool design from these two sets of requirements. Our Mango prototype uses Python 2.7 and Tkinter for the rendering pipeline (Figure 4.3) and UI (Figure 4.4), which communicates with haptic devices via USB.

4.1.2 Framework for Tactile Animation

In this section, we present an animation metaphor that allows users to generate tactile content in the same way as they would create visual animations and play them real-time on a VT array. Figure 4.3 shows the workflow of this authoring mechanism. Designers create tactile animations on a typical animation tool as shown in Figure 4.3a. The animation object is placed in space, and the designer adjusts its size on the visual outline of the VT array. The designer then adds movements and special effects to the object using Mango's toolset, and plays it to observe its frame-by-frame sequence.

Mango's rendering engine translates visual animations to tactile animations on the VT array. Knowing the location of vibrating points on the sparse array of VT actuators, the rendering engine resolves the animated sequence into individual actuators using the phenomena of phantom tactile sensations [1, 34]. The phantom sensation is a sensory illusion elicited by stimulating two or more vibratory elements on the skin. Instead of feeling the individual vibration points, the user feels a single sensation in between, whose perceived intensity is defined by the weighted sum of the intensities of the vibrating elements. Therefore, in each frame, the animated tactile object is resolved into intensity of actuators on the VT array (Figure 4.3b). The rendering engine then calculates raw waveforms for each VT channel (Figure 4.3c) that can either be sent to the VT device to play the animated sequence or exported as a multichannel datafile for later use. Previous work has interpolated between only two actuators [**? ?**]; however, a more generalized 3-actuator interpolation algorithm allows for arbitrary real-time manipulation of the tactile animation object on grid displays.

To accommodate the animation framework, we define three **datatype models**, for use in the current implementation and future expansion of the Mango tool: *Tactile animation objects*, high-level hardware-independent data types for tactile animation; *vector formats*, high-level hardware-specific control common in previous work; and *raster formats*, low-level hardware-specific formats for rendering and playback.

Tactile animation objects are high-level specifications of virtual sensations moving on a 2D VT array (Figure 4.3a). High-level parameters, such as location, size, and other semantic qualities, can either be constant or variable. Each tactile object has a start time and a duration. Object type is also defined for tactile animations that sets pre-defined parameters and features to animated objects. For example, a moving virtual point can have a position, size, and frequency parameter, while a "rain" effect can have a position and more semantic parameters like raindrop frequency or size.

Tactile animation objects are device-independent. Mango uses a device configuration file (LR4) and the rendering engine to create animated VT patterns on hardware. Animation objects can be combined in novel ways, organized in groups, or generate other tactile animations like a particle generator as in a graphical animation tool, and can have paths that constrain motion to a pre-determined trajectory. We prototyped an early version of the tactile animation object in Mango; however, the data type is extensible.

Vector formats are similar to those in previous work (e.g., [20]). Instead of objected-based definitions, as in tactile animation objects, parameters are defined

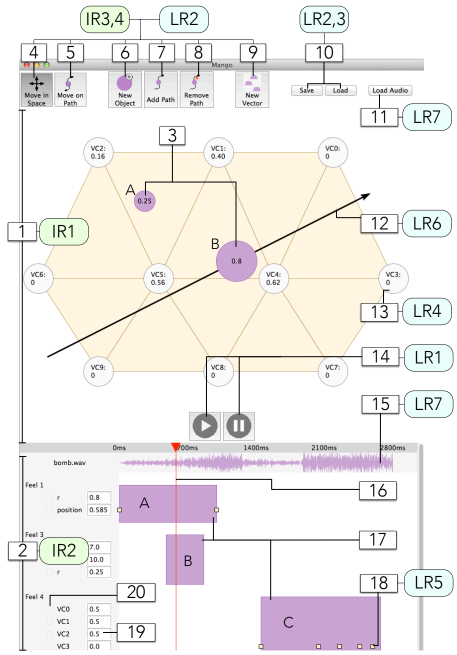


Figure 4.4: Mango graphical user interface. Key components are labeled and linked to corresponding design requirements.

for individual actuation. (Figure 4.3b). Parameters include duration, amplitude envelopes (e.g., fade-ins and fade-outs), frequency, and start times. Being device-specific, vector formats offer finer sensation control than tactile animation objects (analogous to pixel-level editing of sprites). However, creating a single percept from independent controls can be challenging. This data type is useful when rendering methods for the hardware are not defined or the user wants to control specific actuator sequence to animate tactile content, such as using the Tactile Brush [34].

Raster format, analogous to a raster-graphics image or WAV file, is suitable for playback operations or exporting it to a device specific format (Figure 4.3c). A raster format contains a matrix of actuator intensities; each row defines intensities of an actuator and columns containing the intensities at each time instance. Each format also contains a timestamp row defined by the rendering engine's framerate. The playback system parses the raster data, finds the current column, and pushes these actuator settings to the device. This data type is also used for real-time feedback during authoring.

4.1.3 Authoring Interface

The authoring interface allows designers to efficiently create moving tactile content in a familiar environment. Here we describe user interactions, most of which are through the animation window (1) and timeline (2) (Figure 4.4).

Animation Window: A user creates a tactile animation object (3) with a "new object" button (6), then manipulates it in the animation window (1). The window is overlaid with a faint trace of the VT hardware (13) for context. Here, we used an array of 10 VT actuators (Figure 4.6).

Object Paths: The animation object (3A) has (x, y) parameters describing position, an "r" (radius) parameter, corresponding to the VT output voltage from 0 (minimum) to 1 (maximum). An optional path can be added to an object (7), or removed (8), along which the motion of the object (3B) is constrained (12). The path-object (3B) is manipulated in two ways: moving on path (5), which moves the object from the beginning (position=0) to the end of the path (position=1), or moving in space (4), which moves the object and the path together on the animation window (1). The current Mango implementation only supports straight-line paths, however their use can be extended in a later version. Also note that curves can be accomplished through keyframed (x, y) positions.

Timeline: Each animation object (3) is represented in the timeline (2) as a track (17). The red scrubhead (16) (shown as a triangle and line) shows and manipulates the current time. Animation objects can be moved in time by clicking and dragging, and resized to change duration. Individual parameters can be set on the left, by typing values into text fields (19), allowing precision. The entire animation can be played and paused using buttons (14) or the spacebar.

Keyframes: Parameters can be toggled as "keyframeable" with a small clock button (20). When the value is changed, a keyframe (18) is automatically created at the current time. Intermediate values are linearly interpolated.

Vector Sensations: A new vector can be created by selecting an object (3) then clicking on a button (9). These sensations control each actuator directly through the parameter values, controlling that actuator's voltage from 0 to 1 (same as the "r" parameter). The corresponding actuator is highlighted in the animation window (1) when the text field (19) or track (17C) is selected. Each track is also keyframeable.

Save and Load: Animations can be saved and loaded (10) to/from JSON files. An audio track can be loaded (11) to the timeline (15).This allows the user to design a VT experience for sound files (LR7). Video overlay is left for future work.

Hardware Configuration File: A hardware-specific structure is defined and stored in a JSON configuration file (LR4). The file contains: (a) physical width and height of the grid, (b) a dictionary of actuator types (e.g., voice coils or rumble motors), each with a list of control parameters (e.g., frequency, intensity) and allowable values; (c) location and type of each actuator; (d) supported communication protocols and rendering methods; (e) brand information (e.g., USB vendor id and product id) for device recognition; and (f) default settings. Physical dimensions are defined in SI units, e.g., meters, Hz.

Playback: Once the animation of the object is defined, the user can play and stop the animation. During playback, the animation runs in (1) and the corresponding parameters vary in (2). Simultaneously, VT stimulations are activated on the hardware for user feedback. Multiple animation objects and vector sensations can exist simultaneously. Actuators output the sum of all the values generated by objects (described later in the Rendering Algorithm section) and vector sensations.

4.2 Rendering Algorithm

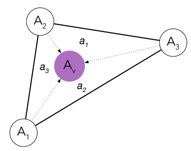
Mango's rendering algorithm defines how high-resolution haptic feedback is translated to sparse grids of VT actuators. The rendering algorithm translates animations created in the animation window to animated VT patterns on the hardware. Figure 4.3 shows the rendering pipeline that converts animation objects to a raster format, which outputs to the hardware.

The rendering algorithm is derived from psychophysical understanding of VT illusions on the skin and creates percepts of virtual actuators and their motion in between a set of real actuators. The precise perceptual model depends on several factors, such as type of VT actuators (DC vs. voice coil motors), stimulation site (forearm vs. back) and the spacing of actuators in the array (e.g., [34]). To allow for custom framerates and real-time feedback, we generalize from the 1D case (in between two VT actuator along a line) to the 2D case (in between three or more actuators, previously accomplished with non-VT sensations [**?**]). Thorough investigation of the psychophysical model is beyond our present scope, however, we empirically determine the most effective model among those documented in the literature for the 1D case with a pairwise comparison.

4.2.1 Perceptual Selection of Interpolation Models

The rendering algorithm translates virtual percepts to a physical actuator grid. We first construct a Delaunay triangulation for all actuators to automatically define a mesh on the hardware grid. At each instant of rendering, we use barycentric coordinates of the virtual animation objects relative to a triangle defined by three real actuators (Figure 4.5a). Barycentric coordinates are scaled by an interpolation method to determine real actuator intensity.

We propose three interpolation models for Mango, derived from prior psychophysical understanding of phantom VT sensations: (i) *linear*, (ii) *logarithmic*



(a) Barycentric coordinates

Linear	$A_i = a_i \times A_v$
Log	$A_i = \frac{\log a_i + 1}{\log A_{max} + 1} A_v$
Power	$A_i = \sqrt{a_i} \times A_v$

(b) Candidate interpolation methods

Figure 4.5: Interpolation models used to determine physical actuator output (A_{1-3}) from virtual actuator intensity (A_v) and barycentric coordinates (a_{1-3}).

("log"), and (iii) *Pacinian power ("power")* (Figure 4.5b).

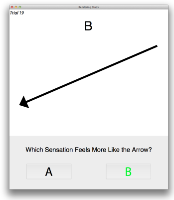
In the linear interpolation model, barycentric coordinates are linearly related to actuation amplitude. In the log model, these coordinates are scaled logarithmically, as perceived intensity is related to physical vibration amplitude [**?**]. In the power model, coordinates are coupled to the power (square of the amplitude) of vibrating stimulations [**?**]. Linear and log interpolation models have been used in the past to express either location or intensity respectively (but not both) of virtual sensations between two vibrators [1**?**]. A Pacinian power model was used in [34] to account for both location and intensity of virtual sensation between two vibrators.

4.2.2 Pairwise Comparison Study

To determine the preferred model for this VT hardware in Mango's rendering pipeline, and to identify relevant factors (e.g., frequency, amplitude), we performed a pairwise comparison of our three candidate interpolation models.

Participants and Apparatus

Eighteen volunteers took part (6 female, between age 20-35). The VT hardware consisted of 10 high-quality VT actuators (C2 tactors, Engineering Acoustics, Inc., USA) arranged in a 3-4-3 layout and mounted on the back of a chair in a pad 21 cm high, 29 cm wide, and 2 cm thick; actuators form equilateral triangles with edges of 6.35 cm (Figure 4.6b). The rendering engine updates at 100 Hz. Through piloting, we determined that the device's on-screen visual outline should mirror the sensations rendered on the physical device. That is, if participants see an animation object on the right side of the screen, they prefer to feel it on the right side of the back. Figure 4.6a shows the experiment interface, in which an arrow represents the



(a) Rendering study interface

(b) Output device with highlighted actuators

Figure 4.6: Rendering study setup and user interface.

sensation direction.

Methods

We conducted A/B paired comparison tests (two-alternative, forced-choice) to determine the preferred model out of the three candidates. In each trial, participants were presented with two stimuli at a 400 ms interval. Each stimulus is a "straight-line" VT stimulation on the back using one model. Participants were asked to select the stimuli that *best represented straight-line motion* in a variety of directions.

Two durations (500 and 1500 ms), eight cardinal directions, and A/B order were crossed with each model pair, and presented in a random order. For each trial, frequency was randomly selected from 80, 160, 240, and 300 Hz, and intensity from between 10 and 20 dB above detection threshold. Each participant performed 96 trials over ∼15min (1728 total).

Results

Each algorithm pair's data was fit to a logistic regression model with participant, frequency, intensity, direction, and duration as factors; direction was grouped into horizontal, vertical, and diagonal. We performed stepwise regression (backwards elimination with $\alpha = 0.05$ and a χ^2 test for removing each factor) to iteratively eliminate factors that were not statistically significant.

Logarithmic vs. Linear. Regression eliminated duration, frequency, intensity, and direction ($p > 0.1$). The resulting model has Nagelkerke $R^2 = 0.135$. Using Bonferroni correction for multiple comparisons, 95% confidence intervals for each participant were computed. 11 participants were more likely to prefer Log over Linear ($p < 0.05$) models; none were likely to prefer the Linear model.

Logarithmic vs. Pacinian power. All 5 factors were eliminated ($p > 0.1$). The overall 95% confidence interval of participants selecting Log over Power was

37.06% to 87.40%, overlapping 50%. We therefore detected no significant difference of preference between Log and Power models.

Pacinian Power vs. Linear. We eliminated intensity, direction and duration ($p > 0.1$), with the fitted model's Nagelkerke $R^2 = 0.0970$. The confidence interval for each participant-frequency combination, via Bonferroni corrections, yielded 22 / 72 participant-frequency combinations selecting Power model over Linear model more than 50% of the time. No one chose the Linear model more than 50% of the time.

Conclusion: Logarithmic interpolation outperformed linear and was equivalent to Pacinian power model. We proceeded with the logarithmic model for Mango's implementation, as the power model did not outperform either of the others.

4.3 Design Evaluation

To evaluate Mango's animation metaphor and expressive capability, we asked media professionals to create a variety of designs. Qualitative evaluation was chosen for rich, focused, early feedback of the animation metaphor and lessons for iteration. A quantitative comparison between tool perspectives is left until more refined tools are developed. We wanted to establish whether this is an effective approach before studying the most effective approach.

Six participants (P1-6, 3 females) were introduced to Mango driving the VT hardware described previously. P1 had experience with haptics but not animation beyond video editing; P2-5 had animation experience but little or no experience with haptics; P6 had no experience with haptics or animation, but was familiar with media tools like Adobe Photoshop. P5 was also involved with the requirement gathering interviews presented earlier. Each entire session took 40 to 60 minutes.

Each participant was introduced to Mango with a training task: designing an alerting sensation using either animation objects or vector sensations (order counterbalanced). Then, each participant was given three design tasks. 1) Primarily *temporal*: create a heartbeat sensation. 2) Primarily *spatial*: tell a driver to turn left. 3) *Context-based*: create a tactile animation to match a sound file. A 3-second sound effect of a bomb falling (with a whistle descending in pitch) then exploding with a boom was chosen, i.e., complex with two semantic components. The wide array of resulting designs can be found in the accompanying video. Mean non-training task time was 5:59 (med 5:38, sd 2:46, range 1:41-13:48).

After each task, participants rated confidence in their design from 1 (Not confident) to 5 (Very confident), primarily to stimulate discussion. All designs were rated 3 or higher; P6 wrote "6" for his sound-based design. The animation object training task was always rated the same or higher than the corresponding vector training task. While suggestive, these ratings were self-reported and from a small sample. We thus did not conduct statistical analysis.

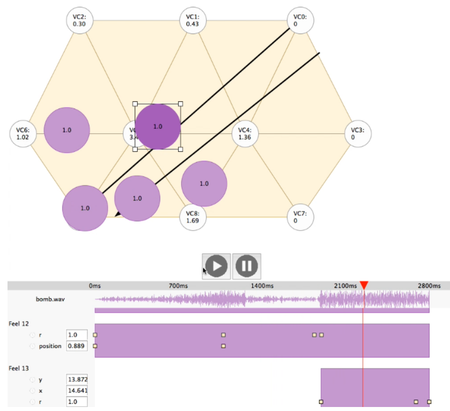


Figure 4.7: Example of P2's animation for matching a sound. See the accompanying video for all participant animations.

A semi-structured interview followed the design tasks. Participants were asked to compare animation objects with vector sensations, and to walk through the interface to elicit feedback. Interviews were conducted and analyzed by a researcher with training and experience in qualitative research, and followed established methodologies: methods of grounded theory [11] informed by phenomenological protocols [52]. Analysis resulted in four themes.

4.3.1 1

Animation Metaphor Participants found the tool easy to use. All six participants were able to accomplish all five tasks (object alert, vector alert, heartbeat, turn left, sound). Participants described the interface as intuitive (P1-5), agreeing that it was an animation tool: *"It's up to the standards of other animation tools"* (P1), *"This is totally animation"* (P2), *"It felt very much like an animation tool"* (P4), *"I'm not an expert when it comes to haptics, but this software seems almost as if it can change the game of designing haptic vibrations"* (P5). Negative feedback focused on polish and feature completeness: *"gotta spline [the keyframe interpolation]"* (P2), *"a couple quirks but there was nothing difficult to overcome"* (P4), *"being able to design your own curve [path] would be really nice"* (P5).

4.3.2 2

Tactile Animation Object vs. Vector Sensations Participants relied more on animation objects than vector sensations, which were only used twice: P4's heartbeat task and P5's sound task (combined with an animation object). P1 switched from vectors to animation objects early in her heartbeat task; no other participants used vector sensations.

Animation objects were described as easier to use and more intuitive, especially to represent location or for non-animators. *"After using the new object I'd probably never use new vector again"* (P2), *"easier to find the location of the heart"* (P1), *"if I weren't an animator I think I would only use [animation objects]"* (P4). Vectors were preferred for more fine-tuned control when motion didn't matter as much, often using many keyframes. *"You can control multiple [actuators] at the same time, so you don't have to create new objects and then put them everywhere on the screen"* (P1), *"[Animation objects] can be more comfortable to use when one doesn't work with keyframes"* (P3), *"If you want precise control over [actuators], then vector is the way to go"* (P4).

4.3.3 3

Designing-in-action with direct manipulation Participants used direct manipulation to feel their designs in real time, dragging animation objects and scrubbing through the timeline: *"I would make the [animation] object and just play around with it before creating the animation, as a way to pre-visualize what I was going to do"* (P5), *"I kind of play around with it, and randomly come up with the ideas"* (P6). P2 even noted that YouTube did not have real-time video scrubbing feedback like Mango's: *"I wish I could scrub back and forth [with YouTube]"* (P2). However, continual vibrations were annoying, and participants requested a "mute" feature: *"It would be nice if...it doesn't go off constantly."* (P3).

More generally, participants used feedback from their experience or external examples. P1 stopped to think about her own heartbeat, P2 used a YouTube video of a heartbeat as a reference, and P3 based her alert on her phone: *"It's typical to have two beeps for mobile phones"* (P3). Correspondingly, participants were excited when prompted by an audio sensation: *"I was really happy with the bomb one, because I could really hear it and imagine me watching a TV and then feel it at the same time"* (P1), *"The sound part was good, that would be a fun thing to design for"* (P4).

4.3.4 4

Replication through Copy and Paste Replication in both space and time was common while using Mango. Many designs had symmetrical paths to reinforce sensations (Figure 4.7). All but P4 requested copy / paste as a feature. *"I could just* *copy/paste the exact same thing on the left side and then move it to the right side"* (P1), *"I have the timing the way I like it, ideally it'd be cool if I was able to copy and paste these, so it would be able to repeat"* (P5).

4.4 Discussion

Here we interpret our design evaluation, explore animation with other devices, and describe applications and limitations.

4.4.1 Design Evaluation Summary

From our design evaluation, we conclude that tactile animation is a promising approach for controlling tactile grids. Direct, continuous manipulation of tactile animation objects supported embodied design and exploration by animators, who rapidly iterated on designs to try new ideas. Mango facilitated the design of a wide variety of animations (see accompanying video) and received positive responses. We also found recommendations for our next iteration: more animation features, video as well as audio context, and muting.

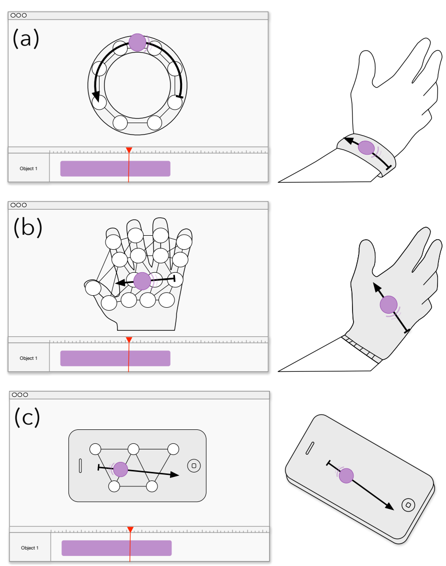


Figure 4.8: Tactile animation could define motion with (a) 1D actuator arrays, (b) dense and sparse VT grids, (c) handhelds.

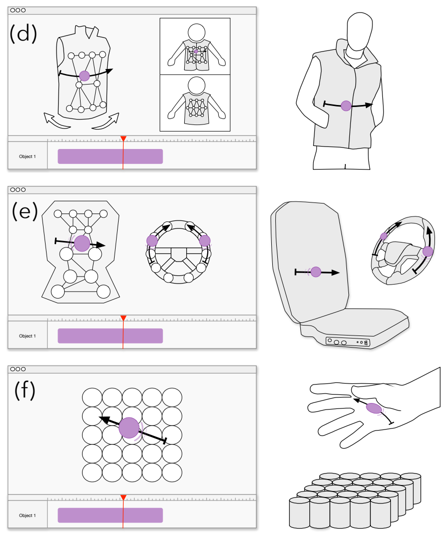


Figure 4.9: Tactile animation could also define motion with (d) 3D surfaces, (e) multi-device contexts, and (f) non-VT devices like mid-air ultrasound.

4.4.2 Possible Extension to Other Device Classes

The animation metaphor is not limited to a back-based pads. Part of the advantage of an abstracted animation object is that, as long as a suitable rendering algorithm can be developed, the metaphor can apply to other devices. In this section, we illustrate possibilities that we plan to explore in future work.

1D VT Arrays (Figure 4.8a): 1D VT arrays are common in arm sleeves, wrist bands, belts, and similar wearables. These devices provide sensations along the path of the array. By constraining objects to a linear or circular path, barycentric coordinates collapse into 1D interpolation.

Dense and Sparse VT Grids (Figure 4.8b): 2D VT grids are also common, used in chairs, gloves, and the backs of vests. While we evaluated Mango with a sparse back-mounted array, tactile animation naturally supports denser arrays, either with our rendering algorithm or by using a nearest-neighbour technique to activate a single actuator.

Handhelds (Figure 4.8c): Actuators embedded in handheld objects, such as mobile devices, game controllers, or steering wheels, shake objects instead of directly stimulating the skin. Animators might be able to define source locations for vibrations using handheld-based rendering algorithms (e.g., [**?**]).

3D Surfaces (Figure 4.9d): Mango currently only supports a 2D location for its animation objects. However, tactile animation can be extended to support surfaces of 3D surfaces, such as vests or jackets that wrap around the user's body. More work will need to be done to perfect this interaction style, possibly using multiple views or a rotatable 3D model with animation objects constrained to the surface.

Multi-device contexts (Figure 4.9e): Mango's rendering algorithm already supports connections to multiple devices simultaneously. The editing interface could combine layouts for different devices, enabling animators to animate the entire user experience (such as a car's seat and steering wheel).

Non-vibrotactile devices (Figure 4.9f): While our rendering algorithm is particular to VT arrays, a tactile animation object can represent manipulable percepts with other actuation technologies. Ultrasound-based mid-air displays generate a sensation as a focal point with a position and size [**?**]; this sensation could be manipulated through a tool like Mango. Similarly, passive force-feedback sensations (e.g., Hapseat [**?**]) or height displays (a grid of pins) could be supported.

4.4.3 Interactive Applications

While our goal was to enable animators to create rich content, the tactile animation object can be linked to alternative input sources for other interactive experiences.

User gestures. User gestures and motion can be tracked and mapped to animation objects directly rendered on the haptic hardware. For example, a user creates patterns on a touch sensitive tablet that maps touch locations to a grid. Users could play games or create personalized haptic messages on the back of a vest. Similarly, a dancer's movements could be tracked through accelerometers, drawing animated haptic content on the body of her audience through actuated theater seats during a live performance.

Camera feed extraction. Motion from video feeds can be automatically extracted with computer vision and rendered on grid displays [**?**], providing dynamic patterns associated with actions during sports, movies, and games. Similarly, animation parameters could be extracted and mapped to positions on a VT grid, creating haptic feedback for non-haptic media.

Data streams. One main application of haptic grid displays is to provide users directional, assistive, and navigational cues during driving cars, walking down the street, or with over- saturated sensory tasks. Users could associate digital data streams, such as GPS input, to predefined set of directional patterns on the back or palm of the hand.

4.4.4 Limitations

While the tactile animation metaphor seems promising and may apply to many contexts, it is limited by the requirement of a suitable rendering algorithm for target hardware. We have not yet explored other form factors, such as handhelds, multi-device scenarios, or non-vibrotactile sensations. Although we perceptually optimized our algorithm, we did not conduct a full psychophysical investigation. Further work needs to be done to identify the limits, thresholds, and peculiarities of this rendering technique. Examples include: curved trajectories of animation objects (although participants' use of curved motion was encouraging, e.g., P5's turn left sensation), spatial frequency control (how to superpose animation objects of differing frequencies), non-triangular meshes (e.g., quadrilateral interpolation or kernel methods), and mixed actuator types (such as a chair with both voice coil and rumble motors, Figure 4.9e).

4.5 Discussion

The Tactile Animation project expanded our understanding from the Haptic Instrument study in Chapter 3. Specifically, it reaffirmed the value of real-time feedback and the need for examples, and showed that a persistent object model reduces cognitive load. It also suggests again that examples and user experiences are extremely valuable to the design process, providing motivation for example-based haptic design discussed next in **??**.

Chapter 5

Combine: Macaron

In both the Haptic Instrument and Tactile Animation case studies, participants drew from their experience or external examples and requested features for repetition. This is unsurprising; creative tasks, like design, are often defined as the recombination of existing ideas, with a twist of novelty or spark of innovation by the individual creator [74]. Examples are critical to provide inspiration, guidance, and inform design [8, 30]; for example, industrial designers collect various knobs and materials, and web designers bookmark sites [30]. Managing these examples effectively is already a significant task even in these more visual fields, but there is no explicit support for vibrotactile (VT) design. I will investigate interaction techniques to directly use examples in haptic design through a design gallery tool for VT icons (**??**).

Design galleries are used in graphics and web design to facilitate the use of examples [42, 48]. While there are several challenges involved with examples in design, including capture, search, management, use, and sharing, I limit this project's scope to the *combining* existing examples to create new VT icons. This project takes place in two phases: Phase I, where we develop a set of tools to manipulate VT icons through interpolation and combination, and Phase II, creating a design gallery and investigating how users work with it (or would like to, if they are unable to do so). For this study, we use a single Haptuator bound to a mobile device to simulate mobile VT icons (**??**). Creativity often sparks when an inventor, examining existing ideas, sees a way to combine them with a novel twist [74]. An environment rich with *examples* is fuel for this fire. In industrial and graphic design [8, 30] their use improves process and final results [17, 42].

Several effect libraries are available to designers of vibrotactile (VT) sensations, e.g., for accessible wayfinding [**?**] or media experiences [15, 35, 63**?**]. But despite the need for effect customizability [67], VT library elements are generally

Figure 5.1: Macaron interface, "`hi`" version featuring both composability (copy and paste), and visibility of underlying parameters. The user edits her sensation on the left, while examples are selected and shown on the right. One editor has focus at a time, shown by the red playhead. Examples are non-modifiable (keyframes cannot be inserted or moved). Macaron is publicly available at `hapticdesign.github.io/macaron`.

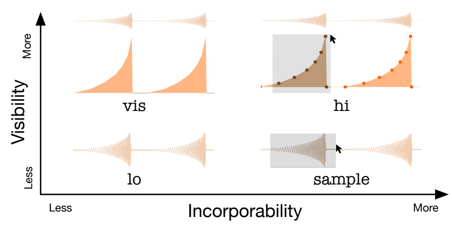


Figure 5.2: Design space for Macaron versions. `hi` and `sample` both allow for selection and copying of example keyframes. `vis` and `hi` both show the underlying profiles. `lo` represents the current status quo; only a waveform is shown.

opaque in construction and immutable. Recent advances include limited parameter adjustability [35, 63] and faceted library search and browsing [68]. Despite this, designers still must either choose a pre-existing sensation or build from scratch: *elements cannot be sampled, recombined, built upon or adapted.* In contrast, web designers can access a page's source; graphic and sound designers can sample and incorporate colours and sounds from other media.

Here, we *examine the potential role of examples* in VT design, to establish how to best support their use. We designed a web-based editor and interactive *design gallery* [42, 48] (Figure 5.1) for VT sensations, then asked users to compare versions (Figure 5.2) that vary in example accessibility via *visibility* and *incorpora-*

bility, as they create VT effects for animations (Figure 5.3).

Analysis of user action logs provide an objective picture of the VT design process. To validate the deployment of this methodology at scale, we also interpret and validate logs with direct observation and interviews. Specifically, we:

- introduce *Macaron*, a web-based VT effect editor through which examples can be used directly in designs,
- find that *visible, incorporable examples make design easier* by providing a starting point for design and scaffolding to learn how to work with VT parameters,
- identify *implications for future tools and libraries*, and
- discuss the *opportunities afforded by a web-based editor* as a practical tool and platform for studying other aspects of VT design at scale.

5.1 Related Work

5.1.1 Salient factors in VT effect perception and control

Vibrotactile effects (e.g. haptic icons [45]) are typically manipulated with low-level engineering of signal parameters, beginning with amplitude, frequency and waveform [3, 26, 45**?**]. Rhythm can support large, learnable icon sets [73**?**]; combining waveforms enhances roughness [**?**]. Time-varying amplitude adds musical expressivity, from tactile crescendos [6] to envelopes [64]. Multi-dimensional scaling can be used to identify and elaborate these parameters [21, 45**? ?**].

Affect and metaphor are another way to structure and manipulate sensations at a level more cognitively relevant than engineering parameters. Perceived valence (pleasantness) and arousal can be influenced by frequency/amplitude combination [**? ?**]. Metaphors [54, 68**?**] and use cases [68**?**] offer structure, memorability and design language. Spatial displays require additional controls for location and direction, whether body-scale [26, 34], mobile [**?**], or mid-air [**?**]. While many parameters are available for VT design, we chose the most established (time-varying frequency and amplitude) for Macaron's initial implementation.

5.1.2 Past approaches to VT design

Past editors – e.g., the Hapticon Editor [20], Haptic Icon Prototyper [71], posVibEditor [62], Vivitouch Studio [72], and Haptic Studio (www.immersion.com) – are track-based, with graphical representations to edit either waveforms or profiles of dynamic parameters. Additional features (e.g., spatial control or mobile interfaces) are surveyed in [**?**].

A library of effects is critical for haptic design tools [**?**]. Most existing tools support feature saving/loading, and some have an internal component library [20, 71, 72]. However, previous implementations were primarily *compositional*, employing building blocks [21] rather than complete artifacts. Example use was not studied.

Large VT libraries contain complete artifacts, but impose a serious constraint on their use. In the Immersion Touch Effects Studio library, underlying structure and design parameters are hidden and cannot be incorporated into new designs. VibViz [68] features 120 VT examples with visualizations searchable by several taxonomies, but the selection model is all-or-nothing. FeelCraft [63] proposes a community-driven library of feel effects [35] for simple parametric customization and re-use. While end user customization-by-selection is important [67], experts need a more open, editable model, just as web designers rely on full access to source code with recent tools allowing search and easy incorporation [42].

5.1.3 Examples in non-haptic design

Problem preparation – also known as the "problem setting" [66] or "analysis of problem" [74] step of design – involves immersion in the challenge and drawing inspiration from previous work. Both may come from the designer's experience, *repertoire* [66] or exposure to a symbolic domain, e.g., mathematical theorems and notation [**?**].

To this end, external examples are critical in inspiring, guiding and informing design [8, 30]. Industrial designers collect objects and materials; web designers bookmark sites [30]. In graphics and web design, *design galleries* organize examples to be immediately at hand [42, 48]. Example-based tools often use sophisticated techniques to mix and match styles and content [**?**]: this requires immediate access to the examples' underlying structure.

5.2 Apparatus Design

To investigate VT design in the context of examples, we required a platform that would expose users' natural procedural tendencies. Our Macaron design gallery is simple, flexible, and extensible. In this work, we add multiple types of example access to polished implementations of familiar concepts: *tracks*, *envelopes*, and *keyframes* (Figures 5.1,5.2).

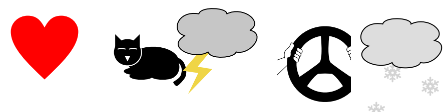
Tracks are the accepted language of temporal media editors (video, audio, and past haptic efforts [20, 62, 71]). We provide tracks for perceptually important

`hi`	Full access to gallery examples, with keyframes visible and selectable for copy and paste. Simulates source visibility, *e.g.*, viewing the source of a web page or having access to a `.psd` PhotoShop document.
`sample`	Hides underlying parameters of frequency and amplitude, whereas waveform regions (underlying keyframes) may be copied and pasted into a design, simulating example mixing in absence of visibility into underlying construction. While possible to see underlying representation by copying the entire example, the steps are indirect and inconvenient.
`vis`	Reveals underlying parameters, but hides keyframes, parameter scales, selection and copy/paste features. The inverse of `sample`, it exposes example structure, but does not support incorporating example elements into a design.
`lo`	Supplies a "black box" outer representation. Playback and visualization of the complete vibration reflect the status quo of non-visible, non-mixable example libraries.
`none`	No examples present.

Table 5.1: Macaron tool alternatives, varied on dimensions of internal visibility and element incorporability.

"textural" parameters (amplitude and frequency); the user accesses periodic and time-variant aspects by manipulating their *envelopes* using *keyframes*, with linear interpolation in-between. Users double-click to create a new keyframe, click or drag a box to select, and change or delete a selection by dragging or with the keyboard. A waveform visualization reflects changes.

Macaron's example access features are inspired by more recent graphics and web design galleries [42, 48**?**], which show examples side-by-side with the editor. Other implemented features, critical for polished creative control [**?**], include real-time playback, time control (scrubbing) copy-and-paste, undo and redo, and



(a) Heartbeat. **(b)** Cat. **(c)** Lightning. **(d)** Car. **(e)** Snow.

Figure 5.3: Animations used as design tasks, in presentation order. Heartbeat expands in two beats; the cat's back expands as breathing and purring; lightning has two arhythmic bolts; the car oscillates up and down, and makes two turns: left then right; snow has three snowflakes float down.

muting (disables realtime VT output). To support its use as an experimental tool, user interactions are logged; start / stop buttons allow the user to indicate when they began and completed their design process.

Macaron was built with HTML5 and JavaScript, using React, Reflux, D3, and Audiolet[1]. Real-time sound synthesis drove a C2 actuator. To leave hands free for keyboard and mouse, the C2 is attached to a wristband; we simulate the design process for a wrist-worn wearable (as in [68]).

Evaluation Versions: To study how examples impact design, we made four gallery versions by sampling two theoretical dimensions of example access: element *incorporability* and internal parameter *visibility* (Figure 5.2, Table 5.1). We hypothesized these would affect users' design processes, e.g., incorporable examples would encourage "mixing and matching" of examples, visibility might provide insight.

We compared these versions with each other and with a non-example version: `none`. In all versions with examples, the user can play or scrub the example, feeling it and seeing the waveform visualization. We did not allow users to modify the examples, to avoid study workflow confounds. To populate the gallery, we chose or adapted seven examples from [68], piloted them to confirm example variety, then regenerated keyframed versions with Macaron.

5.3 Study Methods

Participants were tasked with creating a sensation to accompany five animations (Figure 5.3) – SVGs (scalable vector graphics) which can be played or scrubbed by the same means as navigating Macaron's time control. We chose animation variety (concrete to abstract) and complexity to inspire non-obvious solutions without

[1]facebook.github.io/react, github.com/reflux, d3js.org, github.com/oampo/Audiolet

overwhelming.

Participants were first trained on `none` with no animation, then presented with five animation/version combinations. As the least crucial source of variance, animations were presented in Figure 5.3's constant order, while interface versions were counterbalanced in two 5x5 Latin square designs. Thus, each participant encountered each animation and each interface version once; over all participants, each animation/version combination appeared twice, with Latin squares balancing 1st-order carry-over effects. This design confounds learning with animation task. We believe this is an acceptable tradeoff at this stage, allowing us balance interface order with a single participant session of reasonable length (1-1.5h).

5.4 Results

We targeted a study size of 10 complete participants for a balanced Latin square design, and a manageable sample size for rich, exploratory, qualitative analysis. 13 untrained participants were recruited: P1-10 (7 female, ages 22-35) completed all five tasks, while I1-3 (2 female, ages 29-45) only completed the first three due to time restrictions. Because I1-3 (and P9) all had the same interface order (`lo`, `none`, `vis`, `hi`, `sample`), we suspect that beginning with 'sparse' versions gave insufficient insight into how to design quickly enough to finish the study. I1-3 showed no distinct patterns beyond this; we leave their data for future analysis.

Analysis and Data: A team member trained in qualitative methods analyzed screen recordings, interviews, and logs with grounded theory methods (memoing, open & closed coding [11]) and thematic analysis and clustering [52]. We visualized logs using D3 (Figure 5.4). We chose a qualitative analysis because our goal was to capture the design process, not compare Macaron with previous tools. Our analysis exposed three major qualitative findings, discussed below.

Tool Usability: Overall, the tool was well received, described as *"easy to use"* (P1), *"well made"* (P5), *"pretty neat"* (P9), *"the templates help a lot"* (P3).

Completion time: Overall mean task completion time for P1-10 was 5m48s (median 4m48s, sd 3m52s, min 40s, max 18m23s). We conducted two one-way ANOVAs on completion time; neither interface ($p = 0.87$) nor task ($p = 0.64$) had a significant effect.

5.4.1 Archetypal Design Process

Log visualizations (Figure 5.4) show that users could and did employ Macaron for all key design stages: preparation, initial design, iteration, and refinement. All participants followed this sequence. Some omitted one or more steps depending on personal style and strategies for using examples (below). We list observations

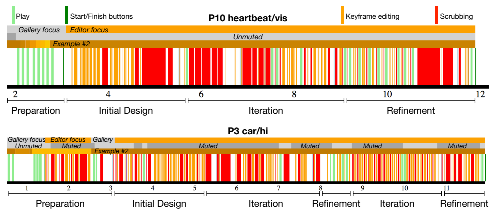


Figure 5.4: Log visualizations showing archetypal design process. Top: P10's heartbeat/`vis` condition (an"ideal" version). Bottom: P3's car/`hi` condition (variations: a return to example browsing after editing, repeated refinement, muted editing).

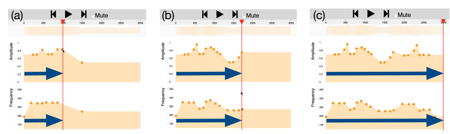


Figure 5.5: P9's cat/`none` design progressed sequentially in time. Note the red playhead helping alignment in (b).

of the basic process in Table 5.2, to document behaviour and frame discussion.

5.4.2 Micro Interaction Patterns Enabled by Tool

Several small-scale patterns further characterize behaviour within the archetypal process.

Different paths through the interface

We saw three design-path strategies.

– *Time* (Figure 5.5; P1,2,3,4,7,9): proceed through the timeline, creating amplitude and frequency at the same time.

– *Component* (Figure 5.6, P1,4,6,8,10): iterate on a design element, then repeat or copy/paste it later in time.

– *Track* (Figure 5.7, P2,3,6,7,8-10): proceed through one entire *track* (typically amplitude), then the other one.

Strategies were often combined hierarchically. P6 developed a car/`lo` component by track (amplitude, then frequency). Wanting additional flexibility, P1,3,7 requested copy/paste *between* tracks: *"The one thing I found missing was copy and pasting between amplitude and frequency"* (P7).

Prepare	All participants began with a problem preparation step [74]. They played the animation to understa problem, then typically looked at several (sometimes all) examples. Only P2, P8, and P9 had a task they did not begin with an example. Otherwise, participants browsed examples, chose a best ma the animation (*"I was trying to find the best match with the visual"* (P7, heartbeat/`hi`)), then tran into initial design. Participants rarely returned to examples for more exploration; only P3 (car/`hi`) (car/`lo`) switched to a different example after beginning their initial design. Preparation is characteri a large number of plays and example switches: on average, 47.45% of all session plays were before t edit (sd 30.15%), and participants switched examples an average of 6.75 times (sd 5.17).
Initial Design	Participants either used their example choice to help create their initial design, or ignored it because it close enough to what they wanted to do. Participants typically recreated the example in their ed copy/paste of the entire design (P1,2,4-8,10) or sometimes a component (P3,10) in incorporable con (`hi` and `sample`), or by manually recreating the design (P5,6) or a component (7,10) with `vis`. In condition, we only observed P5 somewhat recreating an example. Occasionally, participants would c new design loosely based on the example rather than recreating it (P3,4,6-8), when using the *Inspire* ex use strategy (described later).
Iterate	Participants refined designs with longer periods of editing typically book-ended by playing the entire (discussed as "real-time feedback" micro interaction pattern). In some cases, especially when the ex was "close enough", participants skipped iteration (*Adjust* or *Select* example use strategies, described
Refine	Smaller changes forecast design conclusion, e.g., incremental global changes: constant fre (P1,2,5,6,10), alignment (P1,3,6), or pulse height adjustment (P1,3,8,10). This step is sometimes vis activity logs, as most participants (P1,3-10) exhibited more frequent plays of the entire design, and periods of editing/scrubbing. Occasionally, participants repeated larger iterations and refinement (P3 c Figure 5.4).

Table 5.2: Steps in observed archetypal design process.

Further showing diverse workflows, participants requested more powerful controls to work with keyframes as a group, such as widen (P5), reverse (P7), shift everything (P9), move up/down and smooth (P4). Other requested features include looping (P1), hovering over a point to see the value (P1), more detail through a zoomable interface (P4).

Alignment and Copy/Paste are Precise, Convenient

Precision was valued; alignment and copy/paste were used to achieve it. Alignment was sought both in time and to keyframe values. A common technique (Figure 5.5b) was to use the red playhead like a plumb-line to align keyframes with animation features (P1-5,7,9,10) and between the two tracks (amplitude and fre-

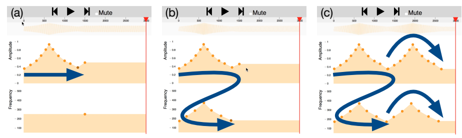


Figure 5.6: P6's car/`lo` design progressed by component, developing the component then repeating it.

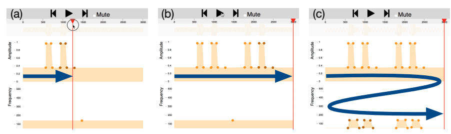


Figure 5.7: P10's heartbeat/`vis` design progressed by track. Amplitude was developed first, then frequency.

quency) (P3-5,7,9,10): *"Using that red arrow thing and placing the dots when it makes the heartbeat"* (P2). Some participants, including those who used the plumb-line, requested more refined alignment features: *"I couldn't keep it straight"* (P1).

Copy/paste was used for improved work efficiency (especially helpful during initial layout or when creating long or repeating designs) and precision: *"Copy and paste...was also the most precise, because if you feel like it's a perfect fit, you can use it exactly"* (P6). Correspondingly, conditions without copy/paste (*i.e.*, `lo` and `vis`) took additional effort: *"It's harder...because there's no copy and paste"* (P5). Precision also depended on context: *"For monitoring someone's health, you would have to be very accurate"* (P9)

Editing and playback

During iteration, participants edited in bursts of primarily scrubbing activity, bookended by full playthroughs. They took time to realize each new version of the design before observing an overview. When editing, participants scrubbed back-and-forth, varying speed (P1-4,7,9,10), and dragging keyframes to try ideas out (P1,3,4,7,9,10) Figure 5.8. This feature was valued by those who used it: *"The real-time part is pretty important"* (P1); some rarely played, showing more frequent or longer periods of scrubbing instead (P2,9,10). Others rarely scrubbed (P5, P8), possibly to have an overall sense of the design: *"Trying to get a general sense of how it might feel"* (P8). P3, P4, and P7 all exhibited focused editing with mute enabled, unmuting for the bookended play sections; others did not use muting.

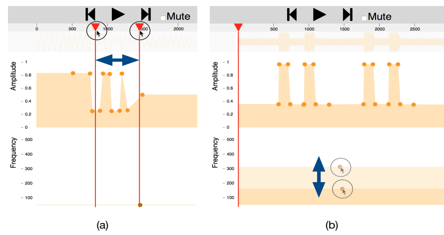


Figure 5.8: Participants used real-time feedback to explore, both (a) in time by scrubbing back and forth (P3 lightning/`lo`), and (b) by moving keyframes (P10 heartbeat/`vis`).

Encoding and Framing

Some participants encoded parameters using consistent rules, often aligned to events like heartbeats or lightning bolts. Others sought to create moods or metaphors for sensation.

Encoding was most visible in the lightning task, where participants represented lightning bolts in regular ways: *"if there was a lightning bolt on the left, I put amplitude and frequency a little longer than a lightning bolt on the right"* (P9). When the animation had two simultaneous bolts, several (P2-4,7,9) encoded it by superimposing two bolt representations on top of one another. Participants were forced to reframe their encoding strategy: *"...two [lightning bolts]...I divided it into two equal partitions, .6 and 1"* (P7).

Encoding failed when participants did not find a direct mapping: *"When the three [snow flakes] come together I think my strategy broke down"* (P7). Metaphors helped in these cases. Car took extra imagination, either for the experience of driving (P6, P8, and P9 didn't drive), or because it's hard to *"know what it would feel like on the wrist"* (P1). P6 describes her process for both lightning and snow as using mood: *"...what I think the mood is...like snow fall, it's kinda like, very gentle and calm"* (P6).

5.4.3 Example Use

As seen, examples played a major role in users' design processes. Analysis revealed the effect of examples to be more nuanced than a one-to-one mapping of the theoretical dimensions of incorporability and visibility. Emergent themes were instead organized on the *role* of examples: as a *direct starting point* for each design; and to *indirectly scaffold learning* throughout a session. The latter was related to additional themes: task difficulty and individual differences.

Ignore	Deliberately do not choose an example, through either lack of match: *"I didn't [find] the examples that I wanted"* (P1); a desire to challenge themselves or be creative: *"I wanted to do my own thing!"* (P9); or difficulty in using the examples.
Inspire	Choose an example, but do not explicitly copy/paste or replicate it in the editor; instead, design based loosely on example parts, sometimes as an adaptation from memory: *"I just tried to remember what the keyframes were like before, and then I modified it"* (P6 car/`lo`).
Template	Choose an initial example, but alter it considerably. In this case, participants use the example to expedite the process.
Adjust	Find an initial example, skipped major iteration and went directly to the refine stage, sometimes because the example was a close match. To enable this, some participants wanted a more powerful manipulation methods, like inverting (P7).
Select	Copy/paste an example (or manually recreate it), then do not modify; sometimes because the example seemed to match: *"...copy and paste, then confirmed it was the same."* (P5)

Table 5.3: Strategies used by participants to directly use examples as a starting point. Ignore and Inspire did not start with copy/paste; Template, Adjust, and Select did, with varying amounts of editing afterwards. When copy/paste was not available, manual re-creation was used as a stand-in.

Direct example use – task starting point

When participants *prepared* for each task by browsing to find a best-match example, then using it as a starting point, they did this with a spectrum of strategies. These strategies, elaborated in Table 5.3, range from Ignore (examples not used) to Select (an example was the final design).

Indirect example use – observe how to design

Over the course of the session, participants used underlying structures of examples to understand how to design VT icons. This was most evident in the `none` or `lo` condition after participants were first exposed to examples: *"I sort of remembered"* (P4 car/`none`). Some explicitly described learning: *"It gave me a general idea of thinking in big shapes rather than little dots"* (P9 lightning/`vis`).

Most participants commented on the difficulty or ease of their task (P1-5, 7-9). Task difficulty was connected learning (*"It's easy...maybe it's more experience"* (P4 snow/`lo`)) and individual differences. Some people were motivated to learn, and challenge themselves; others were not.

Connections between these factors are complex and difficult to unravel with this data. We speculate on the utility of flow theory [**?**] as a useful lens to connect these issues, as it considers creativity, education, and the relationship between perceived challenge and perceived ability. We plan to use it to frame future exploration.

5.5 Discussion

We discuss implications for design, then limitations we hope to progress on with future work.

5.5.1 Implications for Design

Expose example structures for learning

When exposed to examples' underlying structure, participants are able to build their repertoire and learn VT design conventions like *"big shapes"* (P9). Such scaffolding is particularly crucial in an environment where experienced VT designers and training possibilities are rare. Whether through exploratory tool use or structured with online training programs, examples can expand the VT design practices available to novice designers.

Examples as templates

Participants typically copied an example first before iterating and customizing, suggesting a template model of modifiable source documents as a way to expose structure and reduce effort for designers.

Example Recommender

The time participants spent searching for the suitable examples suggests a recommender system could be very valuable. AI techniques might recommend examples similar (or dissimilar) to a source stimuli, as with previous tools in other sensory modalities [42] and VT visualization tools like VibViz [68].

Clarify example context

Participants often repeated gallery searches for each new animation; they needed to compare examples alongside the target graphic. In addition, though our examples were designed independently of our animation tasks, some participants showed confusion about whether they were supposed to match. Clarifying the context for each example, by presenting it either in connection to its original design goal or as a candidate for the participant's current goal, will help participants choose an example.

Hideable examples

Some participants wanted to be individualistic with their designs and actively disliked the most powerful `hi` condition, saying that the `none` condition was cleaner, or that while examples were helpful to learn, they felt "more creative" with fewer examples present. A hideable gallery, which can be opened when needed but kept hidden otherwise, could accommodate user preference. An intelligent gallery could even time example appearances or suggestions to occur at helpful design stages, e.g., by recognizing by activity patterns [17, 74].

Realtime "prefeel" then render

Macaron's real-time feedback supported exploration, with full play-throughs providing an overview or evaluation in-between editing sessions. In addition, P4, who was familiar with haptics, felt that the scrubbing synthesis was "muddy" relative to waveforms pre-rendered with audio tools – a common challenge, noted also by the researchers but deemed suitable for this study. While we hope this technology deficit inspires improved realtime rendering algorithms, it also suggests an explicit workflow compromise. Many video editing and compositing tools show a low-resolution previsualization in design mode; a clip is then fully rendered for playback. For tactile design, coarse, "prefeel" sensations would be synthesized for immediate feedback during a rough design stage, and a high-fidelity rendering generated for less frequent play-throughs. This could help computationally demanding, perceptually-based models or multi-actuator setups (e.g., tactile animation [**?**

] as a prefeel for tactile brush [34]).

Tool flexibility

Macaron was used in very different ways depending on the participant. Some progressed by time, by track, by component, or a combination thereof. Some mirrored frequency and amplitude, using them together, while others used them to express different ideas. This suggests that tools should be flexible and accommodate different strategies; perhaps offering a choice to group by parameters (e.g., [**?**]) or work along parameter tracks (e.g., [71, 72]).

Alignment tools

Participants frequently used the playhead for alignment, finding locations in the video or aligning points between amplitude and frequency. Participants requested using modifier keys to align points (as in other editing tools), or a visualization of events in video. This suggests several features, providing ability to:

– Align comparison sensations from each modality - visual or audio sensation alongside VT.

– Place anchors for attaching a VT sensation (or keyframe within it) to a point in a target visual or audio sensation. This might be automatically assisted, e.g, with video analysis techniques to find scene changes.

– Automatically align keyframes to nearby keyframes, or use a modifier key to constrain or nudge keyframe movement.

Reuse

Copy/paste, especially from a template, speeded design and facilitated otherwise tedious approaches. Several participants made use of element repetition, which had to be re-done upon design re-framing. While copy/paste was helpful, more powerful repetition tools (e.g. looping, and "master templates", as in PowerPoint) would likely find use by many designers.

Automated Encoding

Some participants applied consistent rules in translating an animation to a tactile rendering – e.g., representing left/right lightning bolts differently in the lightning animation, or directly matching amplitude to up-down motion in the car animation. Some of these practices might be automated into generative rules. For example, video analysis could detect up/down motion for a visual object, and translate that automatically to a level for amplitude, similar to how motion trackers can track a moving object and link that to position of an animation; or, a designer might want to specify the mapping. More complex parameterizations could provide a useful tool for expert users, much like how `fmod` allows for parametrized audio in game design.

5.5.2 Limitations & Future work

Limitations in our study suggest future lines of inquiry: following up on additions study factor by deploying online.

Study factors

Our Latin square design allowed qualitative comparison of several gallery variants, but did not have the power for comparative statistical tests between the alternatives. Meanwhile, five design tasks presented in a uniform order did not permit systematic insights into other factors: learning, or task features such as abstractness and complexity. Flow was identified after-the-fact as an important framework for future analysis, but only after our study was designed and data was collected.

Our proposed example-usage dimensions of visibility and incorporability were a useful starting point, but did not line up well with the task processes that people actually used with Macaron. We did see behaviors that aligned well with *learning* and *design-starting* from examples, as well as hints of a more rich and nuanced view of what makes examples useful and in what way.

First, the examples-as-starting-point strategies actually used (Table 5.3) suggest that visibility and incorporability at minimum are not quite right and probably insufficient in dimensionality – there is a concept of edibility regardless of starting point; whereas incorporability could entail editing, but certainly requires an example as a start.

Additionally, observations (including details not reported due to space limits) suggest other factors that influence example use, e.g., *difficulty*, from task, interface and personal confidence and experience; and *task*, from task complexity and abstraction, user strategy, e.g. encoding and metaphor, and user confidence and experience. These hints are far from orthogonal, and will require further research, with focus turned to elements like task abstraction and user background, to disentangle and prioritize.

Online deployment

Triangulation will be helpful in studying factors like difficulty, task abstraction, and user background. In this study, Macaron was deployed and studied locally. We were able to validate the editor's design support and utility of its logging methods, and expose many interesting insights into natural end-user design practices.

Our next plan to answer these questions is to deploy Macaron at a larger scale: online, as a free-to-use design tool for the haptics community, with an initial study in haptics courses. This will allow research *in-situ* with larger, more quantitative, remote-based methods for data collection, triangulated with the less scalable qualitative methods used in-lab. Interaction logs, use statistics, and A/B tests will help us further develop Macaron as a tool for VT design and more generally as a lens for the haptic design process.

5.6 Conclusion

In this paper, we present initial findings from a vibrotactile (VT) design gallery, Macaron. This tool revealed insights both into how examples are used in VT design and implications for other VT design tools. Macaron was implemented using web tools, offering a unique opportunity to follow-up on the design process we observed here, helping designers to create engaging experiences while understanding their craft.

Chapter 6

Share: HapTurk

In modern handheld and wearable devices, vibrotactile (VT) feedback can provide unintrusive, potentially meaningful cues through wearables in on-the-go contexts [?]. With consumer wearables like Pebble and the Apple Watch featuring high-fidelity actuators, VT feedback is becoming standard in more user tools. Today, VT designers seek to provide sensations with various perceptual and emotional connotations to support the growing use cases for VT feedback (everyday apps, games, etc.). Although low-level design guidelines exist and are helpful for addressing perceptual requirements [3, 7, 33, 73?], higher-level concerns and design approaches to increase their usability and information capacity (e.g., a user's desired affective response, or affective or metaphorical interpretation) have only recently received study and are far from solved [35, 37, 54, 55, 67?]. Tactile design thus relies heavily on iteration and user feedback [64]. Despite its importance [67, 68], collecting user feedback on perceptual and emotional (i.e., affective) properties of tactile sensations in small-scale lab studies is undermined by noise due to individual differences (IDs).

In other design domains, crowdsourcing enables collecting feedback at scale. Researchers and designers use platforms like Amazon's Mechanical Turk (`www.mturk.com`) to deploy user studies with large samples, receiving extremely rapid feedback in, e.g., creative text production [?], graphic design [?] and sonic imitations [?].

The problem with crowdsourcing tactile feedback is that the "crowd" can't feel the stimuli. Even when consumer devices have tactors, output quality and intensity is unpredictable and uncontrollable. Sending each user a device is impractical.

What we need are crowd-friendly proxies for test stimuli. Here, we define a *proxy vibration* as a sensation that communicates key characteristics of a source stimulus within a bounded error; a *proxy modality* is the perceptual channel and representation employed. In the new evaluation process thus enabled, the designer translates a sensation of interest into a proxy modality, receives rapid feedback from a crowd-sourcing platform, then interprets that feedback using known error bounds. In this way, designers can receive high-volume, rapid feedback to use in tandem with costly in-lab studies, for example, to guide initial designs or to generalize findings from smaller studies with a larger sample.

To this end, we must first establish feasibility of this approach, with specific goals: **(G1)** Do proxy modalities work? Can they effectively communicate both physical VT properties (e.g., duration), and high-level affective properties (roughness, pleasantness)? **(G2)** Can proxies be deployed remotely? **(G3)** What modalities work, and **(G4)** what obstacles must be overcome to make this approach practical?

This paper describes a proof-of-concept for proxy modalities for tactile crowdsourcing, and identifies challenges throughout the workflow pipeline. We describe and assess two modalities' development, translation process, validation with a test set translation, and MTurk deployment. Our two modalities are a new technique to graphically visualize high-level traits, and the low-fidelity actuators on users' own commodity smartphones. Our test material is a set of 10 VT stimuli designed for a high-fidelity tactile display suitable for wearables (referred to as "high fidelity vibrations"), and perceptually well understood as presented by that type of display (Figure 6.6). We conducted two coupled studies, first validating proxy expressiveness in lab, then establishing correspondence of results in remote deployment. Our contributions are:

- A way to crowdsource tactile sensations (vibration proxies), with a technical proof-of-concept.
- A visualization method that communicates high-level affective features more effectively than the current tactile visualization standard (vibration waveforms).
- Evidence that both proxy modalities can represent high-level affective features, with lessons about which features work best with which modalities.
- Evidence that our proxy modalities are consistently rated in-lab and remotely, with initial lessons for compliance.

6.1 Related Work

We cover work related to VT icons and evaluation methods for VT effects, the current understanding of affective haptics, and work with Mechanical Turk in other modalities.

6.1.1 Existing Evaluation Methods for VT Effects

The haptic community has appropriated or developed many types of user studies to evaluate VT effects and support VT design. These target a variety of objectives:

1) *Perceptibility:* Determine the perceptual threshold or Just Noticeable Difference (JND) of VT parameters. Researchers vary the values of a VT parameter (e.g., frequency) to determine the minimum perceptible change [**? ?**].

2) *Illusions:* Studies investigate effects like masking or apparent motion of VT sensations, useful to expand a haptic designer's palette [28, 34**?**].

3) *Perceptual organization:* Reveal the underlying dimensionality of how humans perceive VT effects (which are generally different than the machine parameters used to generate the stimuli). Multidimensional Scaling (MDS) studies are common, inviting participants compare or group vibrations based on perceived similarity [10, 73**? ? ?**].

4) *Encoding abstract information:* Researchers examine salient and memorable VT parameters (e.g. energy, rhythm) as well as the number of VT icons that people can remember and attribute to an information piece [7, 10, 73**?**].

5) *Assign affect:* Studies investigate the link between affective characteristics of vibrations (e.g., pleasantness, urgency) to their engineering parameters (e.g., frequency, waveform) [73**? ? ?**]. To achieve this, VT researchers commonly design or collect a set of vibrations and ask participants to rate them on a set of qualitative metrics.

6) *Identify language:* Participants describe or annotate tactile stimuli in natural language [10, 25, 54, 68, 73**?**].

7) *Use case support:* Case studies focus on conveying information with VT icons such as collaboration [10], public transit [**?**] and direction [**? ?**], or timing of a presentation [**?**]. In other cases, VT effects are designed for user engagement, for example in games and movies, multimodal storytelling, or art installations [35**?**]. Here, the designers use iterative design and user feedback (qualitative and quantitative with user rating) to refine and ensure effective design.

All of the above studies would benefit from the large number of participants and fast data collection on MTurk. In this paper, we chose our methodology so that the results are informative for a broad range of these studies.

6.1.2 Affective Haptics

VT designers have the challenge of creating perceptually salient icon sets that convey meaningful content. A full range of expressiveness means manipulating not only a vibration's physical characteristics but also its perceptual and emotional properties, and collecting feedback on this. Here, we refer to all these properties as affective characteristics.

Some foundations for affective VT design are in place. Studies on tactile language and affect are establishing a set of perceptual metrics [54, 68]. Guest *et al* collated a large list of emotion and sensation words describing tactile stimuli; then, based on multidimensional scaling of similarity ratings, proposed comfort or pleasantness and arousal as key dimensions for tactile emotion words, and rough/smooth, cold/warm, and wet/dry for sensation [54]. Even so, there is not yet agreement on an affective tactile design language [37].

Recently, Seifi *et al* compiled research on tactile language into five taxonomies for describing vibrations [68]. **1) Physical properties** that can be measured: e.g., duration, energy, tempo or speed, rhythm structure; **2) sensory properties**: roughness, and sensory words from Guest *et al*'s touch dictionary [25]; **3) emotional interpretations**: pleasantness, arousal (urgency), dictionary emotion words [25]; **4) metaphors** provide familiar examples resembling the vibration's feel: heartbeat, insects; **5) usage examples** describe events which a vibration fits: an incoming message or alarm.

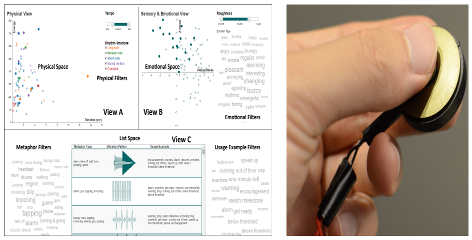
To evaluate our vibration proxies, we derived six metrics from these taxonomies to capture vibrations' physical, sensory and emotional aspects: 1) duration, 2) energy, 3) speed, 4) roughness, 5) pleasantness, and 6) urgency.

6.1.3 Mechanical Turk (MTurk)

MTurk is a platform for receiving feedback from a large number of users, in a short time at a low cost [**?** **?**]. These large, fast, cheap samples have proved useful for many cases including running perceptual studies [**?**], developing taxonomies [**?**], feedback on text [**?**], graphic design [**?**], and sonic imitations [**?**].

Crowdsourced studies have drawbacks. The remote, asynchronous study environment is not controlled; compared to a quiet lab, participants may be subjected to unknown interruptions, and may spend less time on task with more response variability [**?**]. MTurk is not suitable for getting rich, qualitative feedback or following up on performance or strategy [**?**]. Best practices – e.g., simplifying tasks to be confined to a singular activity, or using instructions complemented with example responses – are used to reduce task ambiguity and improve response quality [**?**]. Some participants try to exploit the service for personal profit, exhibiting low task engagement [**?**], and must be pre- or post-screened.

Studies have examined MTurk result validity in other domains. Most relevantly, Heer *et al* [**?**] validated MTurk data for graphical perception experiments (spatial encoding and luminance contrast) by replicating previous perceptual studies on MTurk. Similarly, we compare results of our local user study with an MTurk study to assess viability of running VT studies on MTurk, and collect and examine



(a) VibViz interface [68] **(b)** C2 tactor

Figure 6.1: Source of high-fidelity vibrations and perceptual rating scales.

phone properties in our MTurk deployment.

Need for HapTurk: Our present goal is to give the haptic design community access to crowdsourced evaluation so we can establish modality-specific methodological tradeoffs. There is ample need for huge-sample haptic evaluation. User experience of transmitted sensations must be robust to receiving device diversity. Techniques to broadcast haptic effects to video [40, 49], e.g., with YouTube [**?**] or MPEG7 [**? ?**] now require known high-fidelity devices because of remote device uncertainty; the same applies to social protocols developed for remote use of high-quality vibrations, e.g. in collaborative turn taking [10]. Elsewhere, studies of VT use in consumer devices need larger samples: e.g., perceivability [**?**], encoding of caller parameters [**?**], including caller emotion and physical presence collected from pressure on another handset [**?**], and usability of expressive, customizable VT icons in social messaging [36]. To our knowledge, this is the first attempt to run a haptic study on a crowdsource site and characterize its feasibility and challenges for haptics.

6.2 Sourcing reference vibrations and qualities

We required a set of exemplar source vibrations on which to base our proxy modalities. This set needed to 1) vary in physical, perceptual, and emotional characteristics, 2) represent the variation in a larger source library, and 3) be small enough for experimental feasibility.

6.2.1 High-fidelity reference library

We chose 10 vibrations from a large, freely available library of 120 vibrations (VibViz, [68]), browsable through five descriptive taxonomies, and ratings of taxonomic properties. Vibrations were designed for an Engineering Acoustics C2

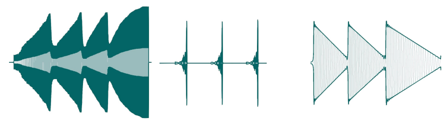


Figure 6.2: VIS_{DIR} Visualization, based on VibViz

tactor, a high-fidelity, wearable-suitable voice coil, commonly used in haptic research [68]. We employed VibViz's filtering tools to sample, ensuring variety and coverage by selecting vibrations at high and low ends of energy / duration dimensions, and filtering by ratings of temporal structure/rhythm, roughness, pleasantness, and urgency. To reduce bias, two researchers independently and iteratively selected a set of 10 items each, which were then merged.

Because VibViz was designed for a C2 tactor, we used a handheld C2 in the present study (Figure 6.1b).

6.2.2 Affective properties and rating scales

To evaluate our proxies, we adapted six rating scales from the tactile literature and new studies. Seifi *et al* [68] proposed five taxonomies for describing vibrations including physical, sensory, emotional, metaphors, and use examples. Three taxonomies comprise quantitative metrics and adjectives; two use descriptive words.

We chose six quantitative metrics from [68] that capture important affective (physical, perceptual, and emotional) VT qualities: 1) *duration* [low-high], 2) *energy* [low-high], 3) *speed* [slow-fast], 4) *roughness* [smooth-rough], 5) *urgency* [relaxed-alarming], and 6) *pleasantness* [unpleasant-pleasant]. A large scale (0-100) allowed us to treat the ratings as continuous variables. To keep trials quick and MTurk-suitable, we did not request open-ended responses or tagging.

6.3 Proxy Choice and Design

The proxies' purpose was to capture high-level traits of source signals. We investigated two proxy channels and approaches, to efficiently establish viability and search for triangulated perspectives on what will work. The most obvious starting points are to 1) visually augment the current standard of a direct trace of $amplitude = f(time)$, and 2) reconstruct vibrations for common-denominator, low-fidelity actuators.

We considered other possibilities (e.g., auditory stimuli, for which MTurk has been used [**?**], or animations). However, our selected modalities balance a) directness of translation (low fidelity could not be excluded); b) signal control (hard

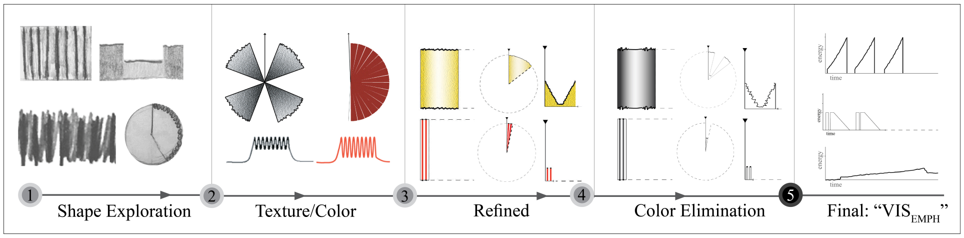


Figure 6.3: Visualization design process. Iterative development and piloting results in the VIS$_{\text{EMPH}}$ visualization pattern.

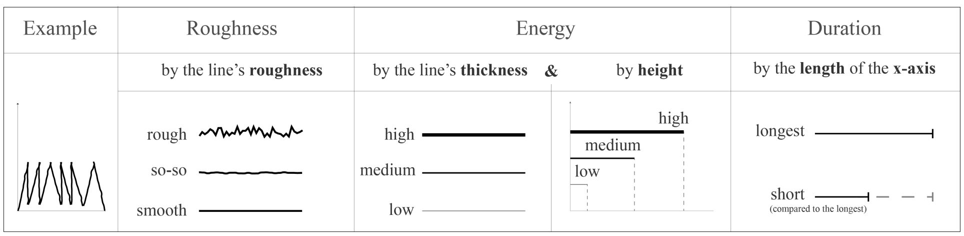


Figure 6.4: Final VIS$_{\text{EMPH}}$ visualization guide, used by researchers to create VIS$_{\text{EMPH}}$ proxy vibrations and provided to participants during VIS$_{\text{EMPH}}$ study conditions.

to ensure consistent audio quality/volume/ambient masking); and c) development progression (visualization underlies animation, and is simpler to design, implement, display). We avoided multisensory combinations at this early stage for clarity of results. Once the key modalities are tested, combinations can be investigated in future work.

“REF” denotes high-fidelity source renderings (C2 tactor).

1) Visual proxies: Norms in published works (e.g. [10]) directed [68] to confirm that users rely on graphical $f(time)$ plots to skim and choose from large libraries. We tested the direct plot, VIS$_{\text{DIR}}$, as the status-quo representation.

However, these unmodified time-series emphasize or mask traits differently than felt vibrations, in particular for higher-level or “meta” responses. We considered many other means of visualizing vibration characteristics, pruned candidates and refined design via piloting to produce a new scheme which explicitly *emphasizes* affective features, VIS$_{\text{EMPH}}$.

2) Low-fidelity vibration proxy: Commodity device (e.g. smartphone) actuators usually have low output capability compared to the C2, in terms of frequency response, loudness range, distortion and parameter independence. Encouraged by expressive rendering of VT sensations with commodity actuation (from early constraints [10] to deliberate design-for-lofi [36]), we altered stimuli to convey high-level parameters under these conditions, hereafter referred to as LOFIVIB.

Translation: Below, we detail first-pass proxy development. In this feasibility stage, we translated proxy vibrations manually and iteratively, as we sought generalizable mappings of the parametric vibration definition to the perceptual quality we wished to highlight in the proxy. We frequently relied on a cycle of user feedback, e.g., to establish the perceived roughness of the original stimuli and proxy candidate.

Automatic translation is an exciting goal. Without it, HapTurk is still useful for gathering large samples; but automation will enable a very rapid create-test cycle. It should be attainable, bootstrapped by the up-scaling of crowdsourcing itself. With a basic process in place, we can use MTurk studies to identify these mappings relatively quickly.

6.3.1 Visualization Design (VIS$_{\text{DIR}}$ and VIS$_{\text{EMPH}}$)

VIS$_{\text{DIR}}$ was based on the original waveform visualization used in VibViz (Figure 6.2). In Matlab, vibration frequency and envelope were encoded to highlight its pattern over time. Since VIS$_{\text{DIR}}$ patterns were detailed, technical and often inscrutable for users without an engineering background, we also developed a more interpretive visual representation, VIS$_{\text{EMPH}}$; and included VIS$_{\text{DIR}}$ as a status-quo baseline.

We took many approaches to depicting vibration high-level properties, with visual elements such as line thickness, shape, texture and colour (Figure 6.3). We first focused on line sharpness, colour intensity, length and texture: graphical waveform smoothness and roughness were mapped to perceived roughness; colour intensity highlighted perceived energy. Duration mapped to length of the graphic, while colour and texture encoded the original's invoked emotion.

Four participants were informally interviewed and asked to feel REF vibrations, describe their reactions, and compare them to several visualization candidates. Participants differed in their responses, and had difficulties in understanding VT emotional characteristics from the graphic (i.e. pleasantness, urgency), and in reading the circular patterns. We simplified the designs, eliminating representation of emotional characteristics (color, texture), while retaining more objective mappings for physical and sensory characteristics.

VIS$_{\text{EMPH}}$ won an informal evaluation of final proxy candidates (n=7), and was captured in a translation guideline (Figure 6.4).

6.3.2 Low Fidelity Vibration Design

For our second proxy modality, we translated REF vibrations into LOFIVIB vibrations. We used a smartphone platform for their built-in commodity-level VT displays, their ubiquity amongst users, and low security concerns for vibration imports to personal devices [**?**]. To distribute vibrations remotely, we used HTML5 Vibration API, implemented on Android phones running compatible web browsers (Google Chrome or Mozilla Firefox).

As with VIS_{EMPH}, we focused on physical properties when developing LOFIVIB(our single low-fi proxy exemplar). We emphasized rhythm structure, an important design parameter [73] and the only direct control parameter of the HTML5 API, which issues vibrations using a series of on/off durations. Simultaneously, we manipulated perceived energy level by adjusting the actuator pulse train on/off ratio, up to the point where the rhythm presentation was compromised. Shorter durations represented a weak-feeling hi-fi signal, while longer durations conveyed intensity in the original. This was most challenging for dynamic intensities or frequencies, such as increasing or decreasing ramps, and long, low-intensity sensations. Here we used a duty-cycle inspired technique, similar to [36], illustrated in Figure 6.5.

To mitigate the effect of different actuators found in smartphones, we limited our investigation to Android OS. While this restricted our participant pool, there was nevertheless no difficulty in quickly collecting data for either study. We designed for two phones representing the largest classes of smartphone actuators: Samsung Galaxy Nexus, which contains a coin-style actuator, and a Sony Xperia Z3 Compact, which uses a pager motor resulting in more subdued, smooth sensations. Though perceptually different, control of both actuator styles are limited to on/off durations. As with VIS_{EMPH}, we developed LOFIVIB vibrations iteratively, first with team feedback, then informal interviews (n=6).

6.4 Study 1: In-lab Proxy Vibration Validation (G1)

We obtained user ratings for the hi-fi source vibrations REFand three proxies (VIS_{DIR}, VIS_{EMPH}, and LOFIVIB). An in-lab format avoided confounds and unknowns due to remote MTurk deployment, addressed in Study 2. Study 1 had two versions: in one, participants rated visual proxies VIS_{DIR} and VIS_{EMPH} next to REF; and in the other, LOFIVIB next to REF. REF_{VIS} and $\text{REF}_{\text{LOFIVIB}}$ denote these two references, each compared with its respective proxy(ies) and thus with its own data. In each substudy, participants rated each REF vibration on 6 scales [0-100] in a computer survey, and again for the proxies. Participants in the visual substudy did this for both VIS_{DIR} and VIS_{EMPH}, then indicated preference for one. Par-

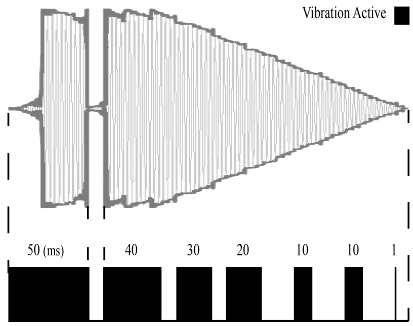


Figure 6.5: Example of LOFIVIB proxy design. Pulse duration was hand-tuned to represent length and intensity, using duty cycle to express dynamics such as ramps and oscillations.

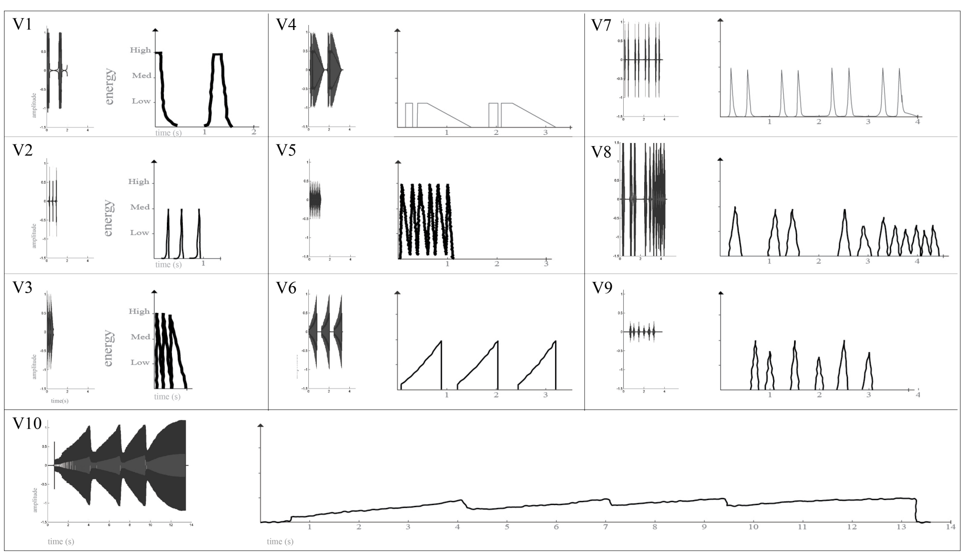


Figure 6.6: Vibrations visualized as both VIS_{DIR} (left of each pair) and VIS_{EMPH}.

ticipants in the lo-fi study completed the LOFIVIB survey on a phone, which also played vibrations using Javascript and HTML5; other survey elements employed a laptop. 40 participants aged 18-50 were recruited via university undergraduate mailing lists. 20 (8F) participated in the visual substudy, and a different 20 (10F) in the low-fi vibration substudy.

Reference and proxies were presented in different random orders. Pilots confirmed that participants did not notice proxy/target linkages, and thus were unlikely to consciously match their ratings between pair elements. REF/proxy presentation order was counterbalanced, as was VIS_{DIR}/VIS_{EMPH}.

6.4.1 Comparison Metric: Equivalence Threshold

To assess whether a proxy modalities were rated similarly to their targets, we employed *equivalence testing*, which tests the hypothesis that sample means are within a threshold δ, against the null of being outside it [**?**]. This tests if two samples are equivalent with a known error bound; it corresponds to creating confidence intervals of means, and examining whether they lie entirely within the range $(-\delta, \delta)$.

We first computed least-squares means for the 6 rating scales for each proxy modality and vibration. 95% confidence intervals (CI) for REF rating means ranged from 14.23 points (Duration ratings) to 20.33 (Speed). Because estimates of the REF "gold standard" mean could not be more precise than these bounds, we set equivalence thresholds for each rating equal to CI width. For example, given the CI for Duration of 14.23, we considered proxy Duration ratings equivalent if the CI for a difference fell completely in the range $(-14.23, 14.23)$. With pooled standard error, this corresponded to the case where two CIs overlap by more than 50%. We also report when a *difference* was detected, through typical hypothesis testing (i.e., where CIs do not overlap).

Thus, each rating set pair could be *equivalent*, *uncertain*, or *different*. Figure 6.8 offers insight into how these levels are reflected in the data given the high rating variance. This approach gives a useful error bound, quantifying the precision tradeoff in using vibration proxies to crowdsource feedback.

6.4.2 Proxy Validation (Study 1) Results and Discussion

Overview of Results

Study 1 results appear graphically in Figure 6.7. To interpret this plot, look for (1) equivalence indicated by bar color, and CI size by bar height (dark green/small are good); (2) rating richness: how much spread, vibration to vibration, within

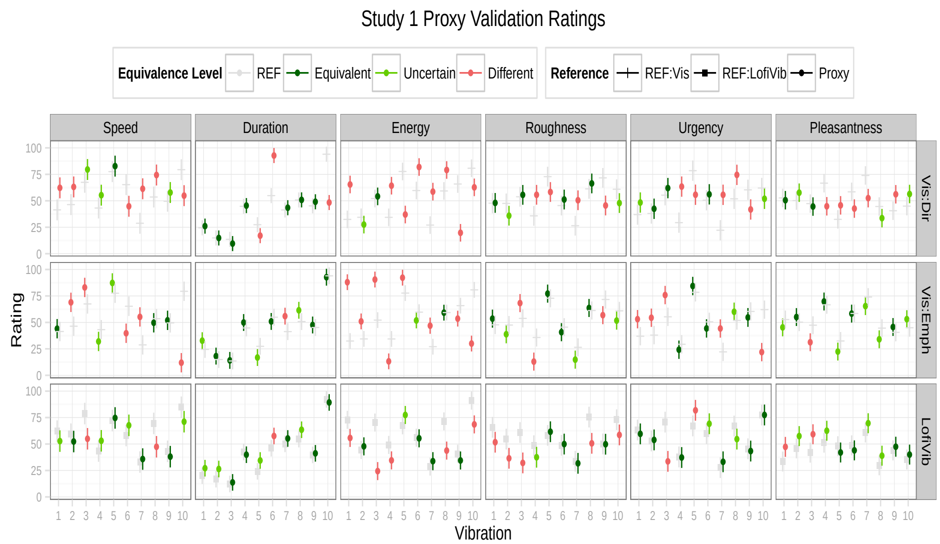


Figure 6.7: 95% confidence intervals and equivalence test results for Study 1 - Proxy Validation. Grey represents REF ratings. Dark green maps equivalence within our defined threshold, and red a statistical difference indicating an introduced bias; light green results are inconclusive. Within each cell, variation of REF ratings means vibrations were rated differently compared to each other, suggesting they have different perceptual features and represent a varied set of source stimuli.

a cell indicates how well that parameter captures the differences users perceived; (3) modality consistency: the degree to which the bars' up/down pattern translates vertically across rows. When similar (and not flat), the proxy translations are being interpreted by users in the same way, providing another level of validation. We structure our discussion around how the three modalities represent the different rating scales. We refer to the number of *equivalents* and *differents* in a given cell as [x:z], with $y =$ number of *uncertains*, and $x+y+z = 10$.

Duration and Pleasantness were translatable

Duration was comparably translatable for LOFIVIB [5:1] and VIS$_{\text{EMPH}}$ [6:1]; VIS$_{\text{DIR}}$ was less consistent [7:3] (two differences very large). Between the three modalities, 9/10 vibrations achieved equivalence with at least one modality. For Duration, this is unsurprising. It is a physical property that is controllable through the Android vibration API, and both visualization methods explicitly present Du-

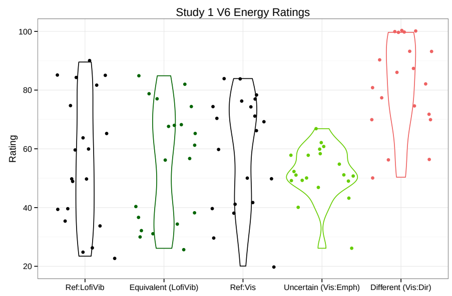


Figure 6.8: Rating distributions from Study 1, using V6 Energy as an example. These violin plots illustrate 1) the large variance in participant ratings, and 2) how equivalence thresholds reflect the data. When equivalent, proxy ratings are visibly similar to REF. When uncertain, ratings follow a distribution with unclear differences. When different, there is a clear shift.

ration as their *x*-axis. This information was apparently not lost in translation.

More surprisingly, Pleasantness fared only slightly worse for LOFIVIB [4:2] and VIS$_{\text{EMPH}}$ [4:1]; 8 / 10 vibrations had at least one modality that provided equivalence. Pleasantness is a higher-level affective feature than Duration. Although not an absolute victory, this result gives evidence that, with improvement, crowdsourcing may be a viable method of feedback for at least one affective parameter.

Speed and Urgency translated better with LOFIVIB

LOFIVIB was effective at representing Urgency [6:2]; VIS$_{\text{EMPH}}$ attained only [4:5], and VIS$_{\text{DIR}}$ [3:5]. Speed was less translatable. LOFIVIB did best at [4:2]; VIS$_{\text{DIR}}$ reached only [1:6], and VIS$_{\text{EMPH}}$ [3:5]. However, the modalities again complemented each other. Of the three, 9/10 vibrations were equivalent at least once for Urgency (V8 was not). Speed had less coverage: 6/10 had equivalencies (V3,4,6,10 did not).

Roughness had mixed results; best with VIS$_{\text{EMPH}}$

Roughness ratings varied heavily by vibration. 7 vibrations had at least one equivalence (V2,4,10 did not). All modalities had 4 equivalencies each: VIS$_{\text{EMPH}}$ [4:3], VIS$_{\text{DIR}}$ [4:4], and LOFIVIB [4:5].

Energy was most challenging

Like Roughness, 7 vibrations had at least one equivalence between modalities (V1,4,10 did not). LOFIVIB [4:5] did best with Energy; VIS_{EMPH} and VIS_{DIR} struggled at [1:8].

Emphasized visualization outperformed direct plot

Though it depended on the vibration, VIS_{EMPH} outperformed VIS_{DIR} for most metrics, having the same or better equivalencies/differences for Speed, Energy, Roughness, Urgency, and Pleasantness. Duration was the only mixed result, as VIS_{DIR} had both more equivalencies and more differences [7:3] versus [6:1] In addition, 16/20 participants (80%) preferred VIS_{EMPH} to VIS_{DIR}. Although not always clear-cut, these comparisons overall indicate that our VIS_{EMPH} visualization method communicated these affective qualities more effectively than the status quo. This supports our approach to emphasized visualization, and motivates the future pursuit of other visualizations.

V4,V10 difficult, V9 easy to translate

While most vibrations had at least one equivalency for 5 rating scales, V4 and V10 only had 3. V4 and V10 had no equivalences at all for Speed, Roughness, and Energy, making them some of the most difficult vibrations to translate. V4's visualization had very straight lines, perhaps downplaying its texture. V10 was by far the longest vibration, at 13.5s (next longest was V8 with 4.4s). Its length may have similarly masked textural features.

V8 was not found to be equivalent for Urgency and Pleasantness. V8 is an extremely irregular vibration, with a varied rhythm and amplitude, and the second longest. This may have made it difficult to glean more intentional qualities like Urgency and Pleasantness. However, it was only found to be different for VIS_{DIR}/Urgency, so we cannot conclude that significant biases exist.

By contrast, V9 was the only vibration that had an equivalency for every rating scale, and in fact could be represented across all ratings with LOFIVIB. V9 was a set of distinct pulses, with no dynamic ramps; it thus may have been well suited to translation to LOFIVIB.

Summary

In general, these results indicate promise, but also need improvement and combination of proxy modalities. Unsurprisingly, participant ratings varied, reducing confidence and increasing the width of confidence intervals (indeed, this is partial

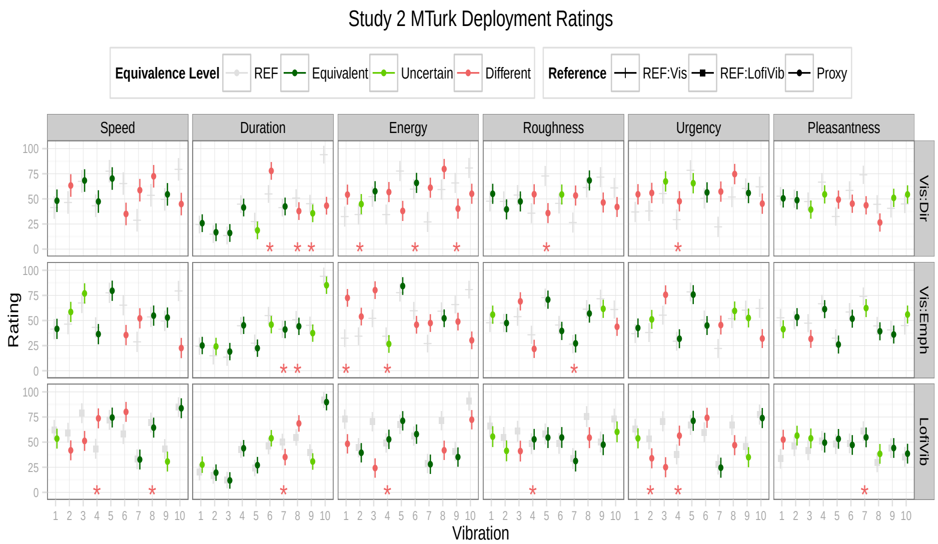


Figure 6.9: 95% Confidence Intervals and Equivalence Test Results for Study 2 - MTurk Deployment Validation. Equivalence is indicated with dark green, difference is indicated with red, and uncertainty with light green. Red star indicates statistically significant difference between remote and local proxy ratings.

motivation to access larger samples). Even so, both differences and equivalencies were found in every rating/proxy modality pairing. Most vibrations were equivalent with at least one modality, suggesting that we might pick an appropriate proxy modality depending on the vibration; we discuss the idea of triangulation in more detail later. Duration and Pleasantness were fairly well represented, Urgency and Speed were captured best by LofiVib, and Roughness was mixed. Energy was particularly difficult to represent with these modalities. We also find that results varied depending on vibration, meaning that more analysis into what makes vibrations easier or more difficult to represent could be helpful.

Though we were able to represent several features using proxy modalities within a bounded error rate, this alone does not mean they are crowdsource-friendly. All results from Study 1 were gathered in-lab, a more controlled environment than over MTurk. We thus ran a second study to validate our proxy modality ratings when deployed remotely.

6.5 Study 2: Deployment Validation with MTurk (G2)

To determine whether rating of a proxy is similar when gathered locally or remotely, we deployed the same computer-run proxy modality surveys on MTurk. We wanted to discover the challenges all through the pipeline for running a VT study on MTurk, including larger variations in phone actuators and experimental conditions (G4). We purposefully did not iterate on our proxy vibrations or survey, despite identifying many ways to improve them, to avoid creating a confound in comparing results of the two studies.

The visualization proxies were run as a single MTurk Human Intelligence Task (HIT), counterbalanced for order; the LOFIVIB survey was deployed as its own HIT. Each HIT was estimated at 30m, for which participants received $2.25 USD. In comparison, Study 1 participants were estimated to take 1 hour and received $10 CAD. We anticipated a discrepancy in average task time due to a lack of direct supervision for the MTurk participants, and expected this to lead to less accurate participant responses, prompting the lower payrate. On average, it took 7m for participants to complete the HIT while local study participants took 30m.

We initially accepted participants of any HIT approval rate to maximize recruitment in a short timeframe. Participants were post-screened to prevent participation in both studies. 49 participants were recruited. No post-screening was used for the visual sub-study. For the LOFIVIB proxy survey, we post-screened to verify device used [**?**]. We asked participants (a) confirm their study completion with an Android device via a survey question (b) detected actual device via FluidSurvey's OS-check feature, and (c) rejected inconsistent samples (eg. 9 used non-Android platforms for LOFIVIB). Of the included data, 20 participants participated each in the visual proxy condition (6F) and the LOFIVIB condition (9F).

For both studies, Study 1's data was used as a "gold standard" that served as a baseline comparison with the more reliable local participant ratings [**?**]. We compared the remote proxy results (from MTurk) to the REF results gathered in Study 1, using the same analysis methods.

6.5.1 Results

Study 2 results appear in Figure 6.9, which compares remotely collected ratings with locally collected ratings for the respective reference (the same reference as for Figure 6.7). It can be read the same way, but adds information. Based an analysis of a different comparison, a red star indicates a statistically significant difference between remote proxy ratings and corresponding local *proxy* ratings. This analysis revealed that ratings for the same proxy gathered remotely and locally disagreed 21 times (stars) out of 180 rating/modality/vibration combination; i.e., relatively infrequently.

Overall, we found similar results and patterns in Study 2 as for Study 1. The two figures show similar up/down rating patterns; the occasional exceptions correspond to red-starred items. Specific results varied, possibly due to statistical noise and rating variance. We draw similar conclusions: that proxy modalities can still be viable when deployed on MTurk, but require further development to be reliable in some cases.

6.6 Discussion

Here we discuss high level implications from our findings and relate them to our study goals (G1-G4 in Introduction).

6.6.1 Proxy modalities are viable for crowdsourcing (G1,G2:feasibility)

Our studies showed that proxy modalities can represent affective qualities of vibrations within reasonably chosen error bounds, depending on the vibration. These results largely translate to deployment on MTurk. Together, these two steps indicate that proxy modalities are be a viable approach to crowdsourcing VT sensations, and can reach a usable state with a bounded design iteration (as outlined in the following sections). This evidence also suggests that we may be able to deploy directly to MTurk for future validation. Our two-step validation was important as a first look at whether ratings shift dramatically; and we saw no indications of bias or overall shift between locally running proxy modalities and remotely deploying them.

6.6.2 Triangulation (G3:promising directions/proxies)

Most vibrations received equivalent ratings for most scales in at least one proxy modality. Using proxy modalities in tandem might help improve response accuracy. For example, V6 could be rendered with LOFIVIB for a pleasantness rating, then as VIS_{EMPH} for Urgency. Alternatively, we might develop an improved proxy vibration by combining modalities - a visualization with an accompanying low-fidelity vibration.

6.6.3 Animate visualizations (G3:promising directions)

Speed and Urgency were not as effectively transmitted with our visualizations as with our vibration. Nor was Duration well portrayed with VIS_{DIR}, which had a shorter time axis than the exaggerated VIS_{EMPH}. It may be more difficult for visual representations to portray time effectively: perhaps it is hard for users to distinguish Speed/Urgency, or the time axis is not at an effective granularity. Animations (e.g., adding a moving line to help indicate speed and urgency), might help to decouple these features. As with triangulation, this might also be accomplished through multimodal proxies which augment a visualization with a time-varying sense using sounds or vibration. Note, however, that Duration was more accurately portrayed by VIS_{EMPH}, suggesting that direct representation of physical features *can* be translated.

6.6.4 Sound could represent Energy (G3:promising directions)

Our high-fidelity reference is a voice-coil actuator, also used in audio applications. Indeed, in initial pilots we played vibration sound files through speakers. Sound is the closest to vibration in the literature, and a vibration signal's sound output is correlated with the vibration energy and sensation.

However, in our pilots, sometimes the vibration sound did not match the sensation; was not audible (low frequency vibrations); or the C2 could only play part of the sound (i.e, the sound was louder than the sensation).

Thus, while the raw sound files are not directly translatable, a sound proxy definitely has potential. It could, for example, supplement where the VIS_{DIR} waveform failed to perform well on any metric (aside from Duration) but a more expressive visual proxy (VIS_{EMPH}) performed better.

6.6.5 Device dependency and need for Energy model for Vibrations (G4:challenges)

Energy did not translate well. This could be a linguistic confusion, but also a failure to translate this feature. For the visualization proxies, it may be a matter of finding the right representation, which we continue to work on.

However, with LOFIVIB, this represents a more fundamental tradeoff due to characteristics of phone actuators, which have less control over energy output than we do with a dedicated and more powerful C2 tactor. The highest vibration energy available in phones is lower than for the C2; this additional power obviously extends expressive range. Furthermore, vibration energy and time are coupled in phone actuators: the less time the actuator is on, the lower the vibration energy. As a result, it is difficult to have a very short pulses with very high energy (V1,V3,V8).

The C2's voice coil technology does not have this duty-cycle derived coupling. Finally, the granularity of the energy dimension is coarser for phone actuators. This results in a tradeoff for designing (for example) a ramp sensation: if you aim for accurate timing, the resulting vibration would have a lower energy (V10). If you match the energy, the vibration will be longer.

Knowing these tradeoffs, designers and researchers can adjust their designs to obtain more accurate results on their intended metric. Perhaps multiple LOFIVIB translations can be developed which maintain different qualities (one optimized on timing and rhythm, the other on energy). In both these cases, accurate models for rendering these features will be essential.

6.6.6 VT affective ratings are generally noisy (G4:challenges)

Taken as a group, participants were not highly consistent among one another when rating these affective studies, whether local or remote. This is in line with previous work [68], and highlights a need to further develop rating scales for affective touch. Larger sample sizes, perhaps gathered through crowdsourcing, may help reduce or characterize this error. Alternatively, it gives support to the need to develop mechanisms for individual customization. If there are "types" of users who do share preferences and interpretations, crowdsourcing can help with this as well.

6.6.7 Response & data quality for MTurk LOFIVIB vibrations (G4:challenges)

When deploying vibrations over MTurk, 8/29 participants (approximately 31%) completed the survey using non-Android based OSes (Mac OS X, Windows 7,8.1, NT) despite these requirements being listed in the HIT and the survey. One participant reported not being able to feel the vibrations despite using an Android phone. This suggests that enforcing a remote survey to be taken on the phone is challenging, and that additional screens are needed to identify participants not on a particular platform. Future work might investigate additional diagnostic tools to ensure that vibrations are being generated, through programmatic screening of platforms, well-worded questions and instructions, and (possibly) ways of detecting vibrations actually being played, perhaps through the microphone or accelerometer).

6.6.8 Automatic translation (G4:challenges)

Our proxy vibrations were developed by hand, to focus on the feasibility of crowdsourcing. However, this additional effort poses a barrier for designers that might negate the benefits of using a platform of MTurk. As this approach becomes better defined, we anticipate automatic translation heuristics for proxy vibrations using validated algorithms. Although these might be challenging to develop for emotional features, physical properties like amplitude, frequency, or measures of energy and roughness would be a suitable first step. Indeed, crowdsourcing itself could be used to create these algorithms, as several candidates could be developed, their proxy vibrations deployed on MTurk, and the most promising algorithms later validated in lab.

6.6.9 Limitations

A potential confound was introduced by VIS_{EMPH} having a longer time axis than VIS_{DIR}: some of VIS_{EMPH}'s improvements could be due to seeing temporal features in higher resolution. This is exacerbated by V10 being notably longer than the next longest vibration, V8 (13.5s vs. 4.4s), further reducing temporal resolution vibrations other than V10.

We presented ratings to participants by-vibration rather than by-rating. Because participants generated all ratings for a single vibration at the same time, it is possible there are correlations between the different metrics. We chose this arrangement because piloting suggested it was less cognitively demanding than presenting metrics separately for each vibration. Future work can help decide whether correlations exist between metrics, and whether these are an artifact of stimulus presentation or an underlying aspect of the touch aesthetic.

Despite MTurk's ability to recruit more participants, we used the same sample size of 40 across both studies. While our proxies seemed viable for remote deployment, there were many unknown factors in MTurk user behaviour at the time of deployment. We could not justify more effort without experiencing these factors firsthand. Thus, we decided to use a minimal sample size for the MTurk study that was statistically comparable to the local studies. In order to justify a larger remote sample size in the future, we believe it is best to iterate the rating scales and to test different sets of candidate modalities.

As discussed, we investigated two proxy modalities in this first examination but look forward to examining others (sound, text, or video) alone or in combination.

6.7 Conclusion

In this paper, we crowdsourced high-level parameter feedback on VT sensations using a new method of *proxy vibrations*. We translated our initial set of high-fidelity vibrations, suitable for wearables or other haptic interactions, into two proxy modalities: a new VT visualization method, and low-fidelity vibrations on phones.

We established the most high-risk aspects of VT proxies, namely feasibility in conveying affective properties, and consistent local and remote deployment with two user studies. Finally, we highlighted promising directions and challenges of VT proxies, to guide future tactile crowdsourcing developments, targeted to empower VT designers with the benefits crowdsourcing brings.

Chapter 7

Applications

The three case studies provide rich but focused data on how to create vibrotactile (VT) experience design tools. To complement these studies, I propose to gather information more broadly to generalize the haptic design process to other use cases and ground in haptician experience (Figure 8.1). In this way, my final contributions will draw from several sources: 1) the three in-depth design studies, 2) insight gathered from several side projects, and 3) two sets of grounded data: interviews with professional designers, and a workshop at World Haptics '15.

Here, I describe the data collection process, then illustrate possible applications and forms for these contributions. However, any resulting theory will be emergent from the data, and can take many forms. To accomplish this in a principled way, I plan to use *memoing* and *constant comparison* [11], looking for common threads between data and double-checking conclusions as new theoretical developments appear. This theory will also draw from a literature review of design theory, summarized in this document's related work section.

7.1 FeelCraft

In recent years, haptic feedback has shown promise to enhance user experience in movies, games, rides, virtual simulations, and social and educational media [1?3]. However, current mainstream media has yet to use the richness of haptic modality within its content. The lack of haptic authoring tools, production infrastructure, standardized playback protocols, and skilled and trained workers has contributed to the difficulty of integrating haptic content with accompanying media. To reduce the gap between haptics and mainstream communication, haptic feedback must be expressive, coherent, and synchronized with the content of the media, and also meet user expectations. We believe that allowing end users to access, customize, and share haptic media will create an intimate, engaging, and personalized experience, and proliferate the use of haptics. In this paper, we propose and implement an architecture that channels media content to dynamic and expressive tactile sensations. We introduce FeelCraft, a media plugin that monitors events and activities in the media, and associates them to user-defined haptic content in a seamless, structured way. The FeelCraft plugin allows novice users to generate, recall, save, and share haptic content, and play and broadcast them to other users to feel the same haptic experience, without having any skill in haptic content generation. In the current implementation, we concentrate on the vibrotactile array as the source of sensation and the back as the surface for stimulation; however, the FeelCraft architecture can be easily adapted for other haptic feedback modalities. We begin by presenting relevant background work related to haptic media infrastructure. We then present the FeelCraft architecture and describe in detail each component of the system. Finally, we conclude the paper with our envisioned application ecosystem.

7.1.1 Related Work

Infrastructures to integrate haptic feedback in media have been primarily derived from media type and user interactions associated with the content. These infrastructures fall into two categories: event triggers and direct mappings. In the event triggers scheme, haptic information is embedded in the media and played back using predefined protocols [4]. A common example is a video game controller that rumbles on predefined triggers embedded in the games. This tech- nique is complex and requires a proper production infrastructure that is expensive. Another disadvantage of this technique is that media designers would require access to tools and libraries for creating expressive haptic effects, similar to the libraries for visual and sound effects [5]. Direct mappings use cues from existing media and directly maps them to haptic effects. For example, a typical way to enhance movies and other visual content with haptics is to monitor activity in a video feed [6] and map these activities to haptic transducers arranged along the seat [1, 7]. This way, movements in a visual scene are mapped to gross motion collocated with events seen in 4D movies and rides (www.d-box.com). Similarly, sound has been used to derive haptic cues for video games and music [8, 9]. For example, Buttkicker technology (Guitammer, USA) shakes the entire seat using low-pass filtered sound. The Vybe Haptic Gaming Pad (Comfort Research) divides sound into three bands and maps them to transducers located in the seat and back. The advantage of the direct mapping scheme is that no change is required in the current media production process. However, each tech- nique is limited to its media type.

A FeelCraft plugin maps media to haptic sensations in a modular fashion, supporting arbitrary media types and output devices. By using a FeelCraft plugin, users can:

- link existing and new media to the haptic feedback technology,
- use an FE library to find appropriate semantically defined effects,
- author, customize, and share a common, evolving repository of FEs, and
- play and broadcast haptic experiences to one or more user(s).

7.1.2 FeelCraft Plugin and Architecture

A pictorial description of the FeelCraft architecture is shown in Fig. 1 archi- tecture. The conceptual framework of FeelCraft revolves around the FE library introduced in [17]. The FE library provides a structured and semantically correct association of media events with haptic feedback. By using the authoring interface to tailor FE parameters, a repository of FEs can remain general while being used for unique, engaging, and suitable sensations for different media. The playback system, authoring and control interface, Event2Haptic mappings, and media plugin support seamless flow of the media content to the haptic feedback hardware.

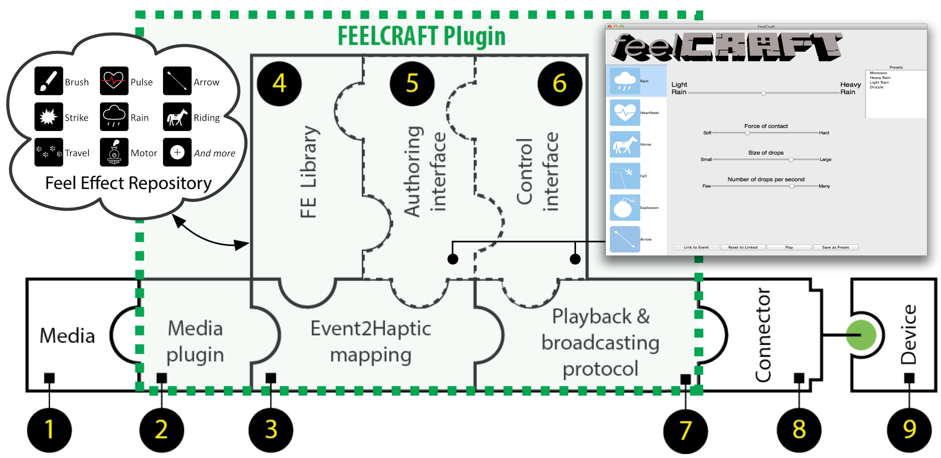


Figure 7.1: FeelCraft architecture. The FeelCraft plugin is highlighted in green. The FE library can connect to shared feel effect repositories to download or upload new FEs. A screenshot of our combined authoring and control interface is on the right.

Media (1) can be entertaining, such as video games, movies, and music, or social and educational. The media can also be general user activity or embedded events in applications. In our implementation, we use the popular sandbox indie game Minecraft (https://minecraft.net).

Media Plugin (2) is a software plugin that communicates with the media and outputs events and activities. This plugin can be as simple as receiving messages from the media or as complicated as extracting events and activities from a sound stream. With existing media, common plugin systems are automatic capture of semantic content from video frames [6], camera angles [1], or sounds [8, 9], or the interception of input devices (such as game controllers or keyboard events). We use a CraftBukkit Minecraft server modification to capture in-game events.

Event2Haptic (3) mappings associate events to FEs, which are designed, tuned, and approved by users using the FE library. This critical component links the media plugin?s output to the haptic playback system. Currently, six FEs are triggered by six recurring in-game events: the presence of rain, low player health, movement on horse, strike from a projectile, in-game explosions, and player falls. Our imple- mentation provides the option to store this mapping directly in the source code, or in a text-based JavaScript Object Notation (JSON) file

FE Library (4) is a collection of FEs. A key feature of an FE is that it correlates the semantic interpretation of an event with the parametric composition of the sensation in terms of physical variables, such as intensity, duration, and temporal onsets [17]. Each FE is associated with a family, and semantically, similar FEs are associated with the same family. For example, the Rain family contains FEs of light rain and heavy rain; as well that that of sprinkle, drizzle, downpour, and rain.In our implementation, each FE family is represented as a Python source file that defines parametric composition of the FE and playback sequences for the FeelCraft Playback system, and each FE is coded as preset parameters in a JSON file. FE family files are necessary to play corresponding FEs in the family, and new FE families can be developed or downloaded through the shared FE repository. The FE can also be created, stored, and shared. FE family and FE files are stored in a local directory of the plugin and loaded into FeelCraft on startup.

Authoring and Control Interfaces (5, 6) allow users to create and save new FEs and tune, edit, and play back existing FEs. Users modify an FE by varying sliders labeled as common language phrases instead of parameters such as duration and intensity (Fig. 1). Therefore, users can design and alter FEs by only using the semantic logic defining the event. The interface also allows users to map game events to new FEs and broadcast to other users, supporting a What-You-Feel- Is-What-I-Feel (WYFIWIF) interface [15].

Playback and Communication Protocols (7) render FEs using the structure defined in FE family files and outputs them through a communication method (8) to one or more devices (9). Our implementation includes an API controlling the commercially available Vybe Haptic Gaming Pad via USB.

7.1.3 Application Ecosystem

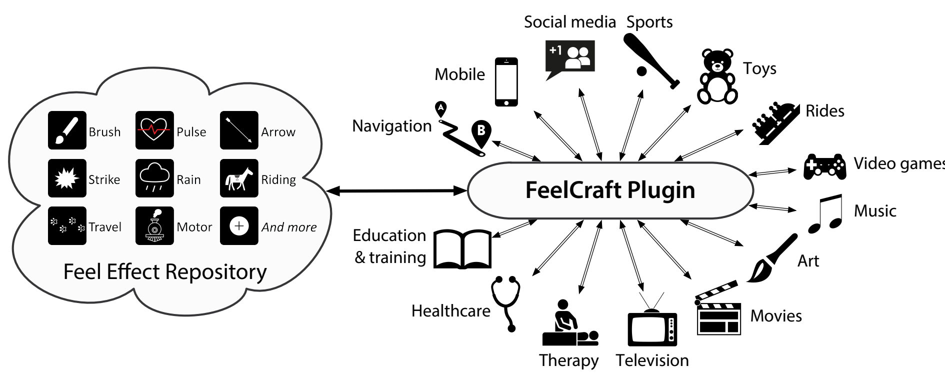


Figure 7.2: Application ecosystem for FeelCraft and an FE repository

FeelCraft plugins are designed to make haptics accessible to end users using existing media and technology. For example, a user may want to assign a custom vibration to a friend?s phone number, or add haptics to a game. In this case, a user would download a FeelCraft plugin for their device, browse FEs on an online feel repository, and download FE families they prefer. Once downloaded, the FeelCraft authoring inter- face allows for customization, as a rain FE for one video game may not quite suit another game. The user could create a new FE for their specific application, and once they were happy with it, upload their customFE for others to use. If the user wanted to show a friend their FE, they could use the playback system to drive output to multiple devices, or export the FE to a file and send it to them later. Figure 2 illustrates this ecosystem with application areas. Just like the Noun Project for visual icons (http:// thenounproject.com) and downloadable sound effect libraries, we envision online repositories of FEs that can be continually expanded with new FEs by users. Our current FErepository includes six original families described in [17] and an additional four new families: Ride, Explosion, Fall, and Arrow.

7.1.4 Conclusion

In this paper, we have presented FeelCraft, a software plugin that allows users to author, customize, share, and broadcast dynamic tactile experiences with media and user activities. The plugin uses FEs, semantically structured haptic patterns. We integrated the FeelCraft plugin with a popular sandbox game, MineCraft. Users can associate six events in the game to corresponding FE, and modify and broadcast to other users to share the haptic experience. The newly authored FEs are saved and shared with other users for communal use via an online FE repository. The pro- posed plugin can also be used with a wide range of entertaining, social, and edu- cational media. Future work includes expanding the FE repository, networked communication and sharing, and supporting output to different haptic device types while maintaining semantic meaning, connecting end users to haptics in an even more accessible manner.

7.2 Feel Messenger

Touch is particularly significant in interpersonal communication [3]. Ranging from mutual user interactions, such as handshakes and hugs, to directing a user's attention by poking or tapping on the shoulder; the channel of touch has also been used in children games in which one child draws patterns on another child's body usin ga finger, and the recipient child guesses what is drawn on the body [7]. Many authors have explored the haptic channel to artificially communicate, enhance and augment interactions on custom-made handheld and wearable devices [3, 4, 5, 16]. This paper explores the use of haptic feedback in social interactions, such as instant messaging, social networking and everyday social use on embedded mobile devices.

Current consumer devices, such as mobile devices, watches and hand controllers, are equipped with a single vibrotactile (VT) motor that usually alerts users for incoming calls or messages, and/or for text entry and ?clicks? [8]. These actuators are popular due to low cost and power requirements, small size and weight, convenient packaging and simple control electronics. The challenge is to create meaningful, expressive and interactive haptic feedback in-between users using only a single low-bandwidth VT motor and with limited computational resources in embedded devices.

Recently, Israr and colleagues [9] have demonstrated that a class of haptic patterns, called feel effects (FEs), could be defined in a parametric structure, called a family. By tuning parameters of these families, users control the semantic interpretation of haptic patterns using the same logic as they would use in normal language. This interpretation of haptic patterns not only allows users to communicate more expressive haptic content, but also to personalize the right haptic pattern for social and interpersonal communication. Such feel effects have also shown to improve early reading comprehension scores in [18].

In this paper, we implement Feel Messenger, a social and instant messaging (IM) application that allows users to share, express and communicate haptic patterns in a structured and economical way on current consumer devices (i.e., using the embedded VT motor). We introduce ?feelbits? and ?feelgits?, haptic structures that allow succinct and eloquent network communication and reduce bandwidth and latency between devices. By allowing users to share and personalize haptic content, we hope to extend the capabilities of handheld and wearable devices for more interactive use among consumers.

This paper is organized as follows: After a brief background, we present the Feel Messenger application. Next, we explore the vocabulary of haptic patterns and present a pilot study to evaluate the usability of the application.

7.2.1 Background

The idea of haptic feedback for instant messaging and social networking has been demonstrated in the past, however, previous work lacked expressive and meaningful haptic interactions connected to a semantic language. In these studies, user gestures, pre- programmed tactile patterns and ?feels? were transmitted using custom devices and through multiple actuators [2, 11-14, 17].

Force feedback devices are not applicable for mobile applications, as they need to be coupled with the ground. Shape-changing features and vibratory grids have been explored to convey emotions, textures and directional cues on handheld devices [6, 8]. A few commercial libraries and APIs are also available to add haptics experiences in current mobile devices [1].

These libraries allow programmers to select from a predefined list of haptic patterns and embed them in their application. All patterns are stored in the device?s memory and users have limited flexibility in personalizing haptic patterns for rich and expressive communication. Recently, Seifi and colleagues [15] have shown that participants preferred customization of VT patterns in social settings, and did not prefer predefined patterns.

Therefore, we present a compact and expressive haptic authoring and sharing protocol and implement it on a typical mobile device. The protocol can be extended for other social networking apps, such as twitter, emails, etc., running on tablets, watches, controllers and other embedded devices.

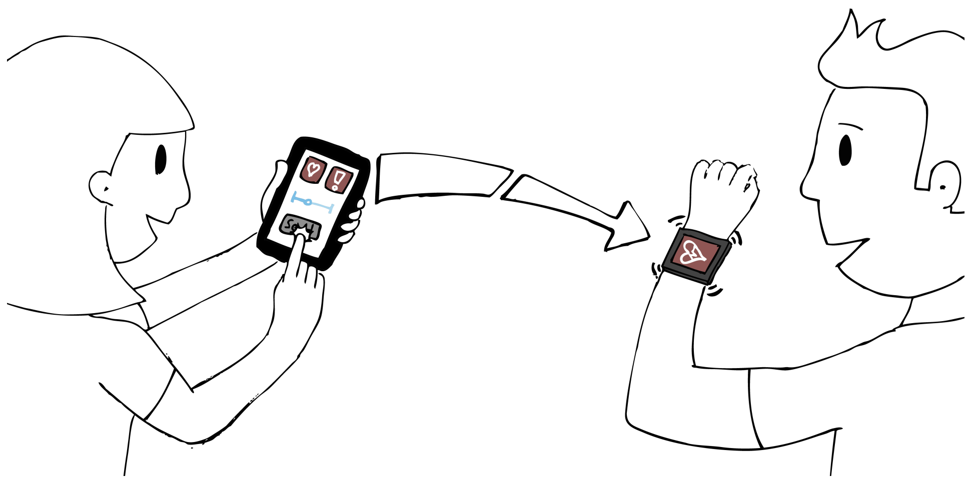


Figure 7.3: Users exchanging expressive haptic messages on consumer embedded devices.

7.2.2 Feel Messenger Application

In this section, we present the architecture (backend) and user interface (frontend) of a messenger application that allows users to create and share haptic content through a network connection. The critical components of the application are shown in Figure 2.

Architecture To account for the limited computation, storage and communication capabilities of a simple microcontroller unit, we introduce feelbits and feelgits. Feelgits (short from ?feel widgets?) are installed piece of software that define parametric compositions of a set of haptic patterns (called a family). Feelbits are parametric settings of a feelgit to produce a particular haptic pattern (called a feel effect [9]). For example, the feelgit of pulse is defined as two successive onsets of vibration, separated by a timing parameter. The feelbits are timing and intensity of onset parameters. Therefore, by varying feelbits, a user can personalize the haptic effect to be calm (low intensity, long temporal separation) or racing (high intensity, short temporal separation) heartbeat.

A library of haptic patterns is stored as parametric models (feelgits) with preset parameters (feelbits). New feelgits and feelbits can be downloaded, personalized and saved. The haptic engine idly waits for incoming haptic messages and renders haptic patterns on demand. Once the message is received, the corresponding feelgit is executed with parameters defined as feelbits. Once the pattern is completely rendered, the engine waits idly for the next message.

Additionally, the response characteristics of the VT motor are also stored in the memory. These characteristics are generally represented by simple first-order functions relating the digital value (such as data byte) to the perceived intensity judged by users [10], which could be used to maintain the quality of experience across wide variety of mobile phones and hardware technologies.

Finally, we introduce a communication protocol that shares feelgits and feelbits along with text messages. For example, the frontend application sends a function playpattern(?pulse?, p1, p2) to play the feelgit pulse with parameters defined as feelbits p1 and p2; or playpulse(?soft?) plays a predefine soft pulse. Note that in order for the device to play a haptic pattern, the corresponding feelgit must be stored in the device; the communication packet includes feelbits and the name or id of the corresponding feelgit.

User Interface The frontend of the application is shown in Figure 2. In addition to conventional text and image entries, the Feel Messenger interface allows users to play, personalize and share haptic patterns. The components of the interface are explained below.

Messenger Interface The message interface, Figure 2 (1), is kept almost the same as a typical IM interface on current devices. However, this can be easily adapted to emails, Twitter feeds or other social media timelines. Aligned to the left are received messages and those aligned to the right are the sent or entered messages. Along with the message is an icon indicating that a haptic effect has been attached to the message. Once the message is received, the application triggers the haptic engine to play the haptic pattern. The user can replay the message by clicking on the message or the accompanying icon. Multiple patterns can also be embedded in a single message. Simultaneous effects are sent by using ?—? (vertical bar) between two icons, otherwise the effects are played in the same order as the icons in the message.

Below the message interface are three icons buttons, shown in Figure 2 (2). The leftmost icon allows users to attach a predefined FE (Figure 3A). The center icon opens a Feel Editor display (Figure 3B), and the right most buttons connects to the users (Figure 3C).

Predefined Patterns

The predefined patterns (Figure 3A) allow users to quickly attach a haptic pattern to the IM. These patterns can be stored from incoming messages or created by using stored haptic families. Each pattern is defined by a set of feelbits that plays when the corresponding feelgit is executed. These presets can be shown as text, images or emotion icons.

Authoring interface

The Feel Editor displays available FE families (feelgits) and allows users to personalize, play, save and share haptic patterns (Figure 3B). By clicking a FE icon, sliders corresponding to parameters (feelbits) are enabled. These sliders may

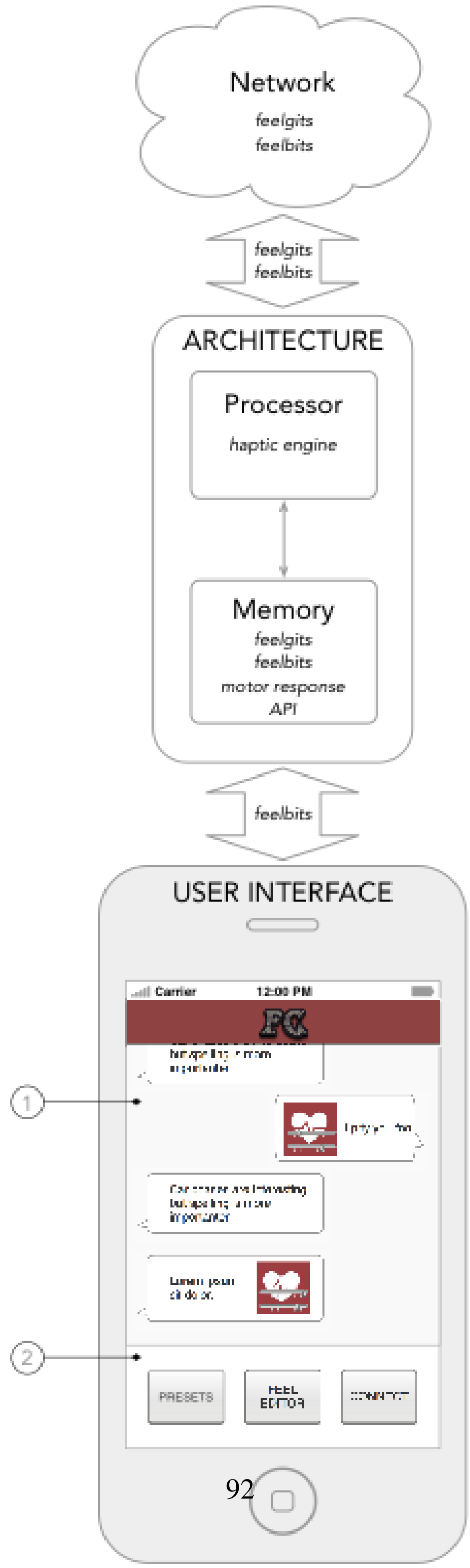


Figure 7.4: The backend software architecture and the frontend user interface of the Feel Messenger app.

have labels corresponding to physical parameters, such as amplitude or duration of vibration; however, we have used semantic labeling that may correspond to single or multiple parameters. Once the sliders are adjusted, the user can play, save or attach the haptic pattern to the IM.

7.2.3 Haptic Vocabulary

The vocabulary of haptic effect is critical for expressive and precise communication between users. In this preliminary implementation, we explore three types of haptic vocabularies. Type 1 is adapted from feel effects defined in [9], where haptic patterns are semantically characterize by a phrase. Type 2 is change in physical parameters as in [4, 12] but can also be simultaneously played with feel effects. Type 3 predefined coded patterns. Figure 4 shows the icons for haptic language. Note, that the two feel effects cannot be simultaneously played. This will result is overflow of the user?s bandwidth, especially with a low-fidelity VT actuator.

Type 1: Feel Effects

A set of feel effects is defined that delivers emotional, attentional and contextual effects. They are:

Pulse: Two successive onsets of vibration; speed (slow/fast) and intensity (weak/strong). Used as pulsation and heartbeat (calm/racing).

Motor: A 4-second modulated vibration; intensity (soft/loud) and speed (slow/-fast) are parameters. Used as snoring, breathing, purring, engine rumble, etc.

Strike: A single onset of vibration; duration (short/long) and intensity of vibration are parameters. Used for tap, poke, jab and punch.

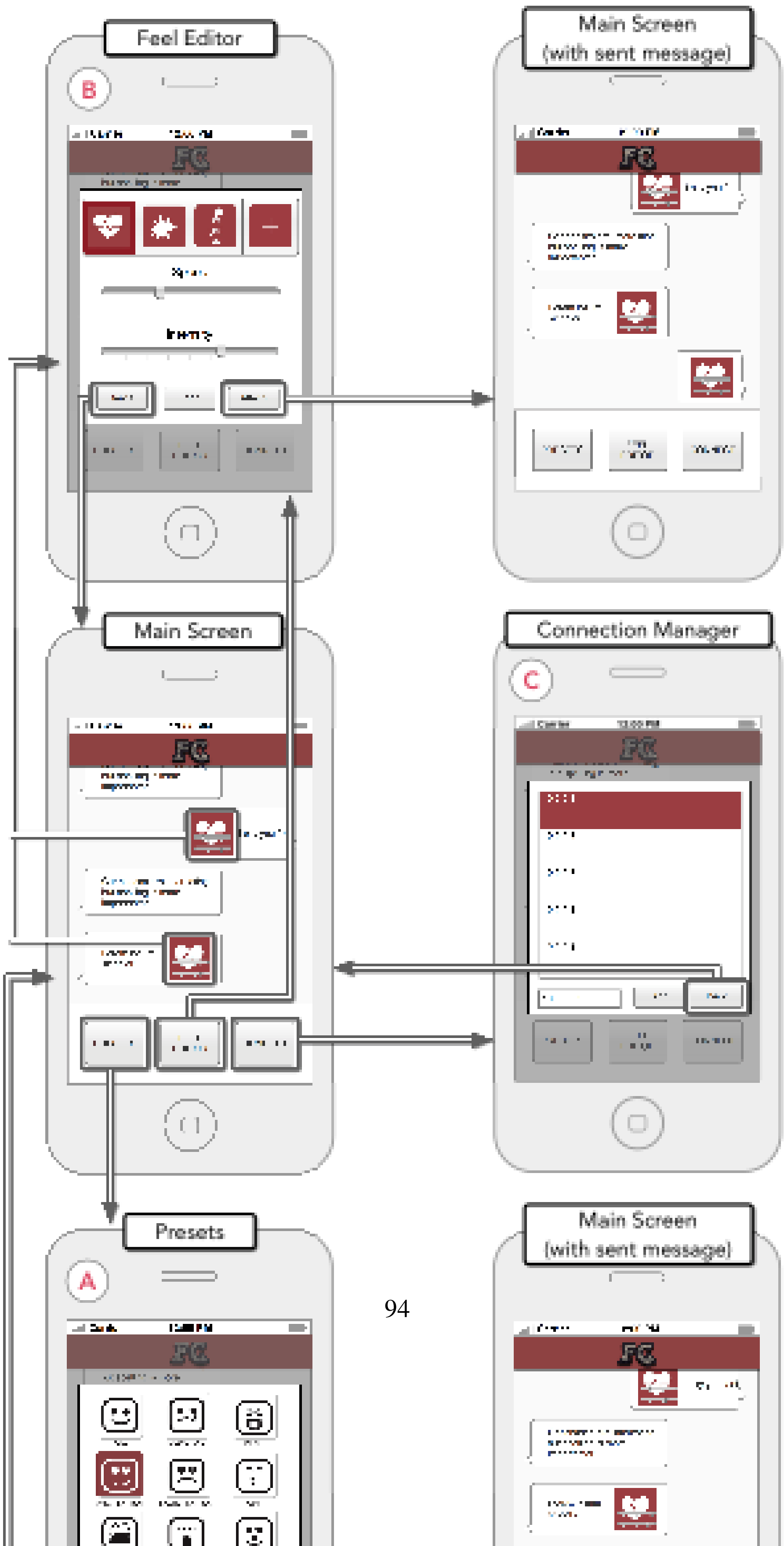
Urgency: a burst of vibrations; intensity (weak/strong) and temporal separation between pulses (low/high urgency) are parameters. Used for alerting users and expressing urgency.

Type 2: Physical Effects

These effects are associated with direct variation in tactile patterns. Previous studies (e.g. [1, 4, 12]) have used variation in amplitude, duration as typical variation. Our library includes:

Ramp-up: gradual increase in intensity; parameters are peak intensity and the rate of increase.

Ramp-down: gradual decrease in intensity; parameters are peak intensity and the rate of decrease.


Feel Editor
B
Main Screen
(with sent message)
Main Screen
Connection Manager
C
Presets
A
Main Screen
(with sent message)

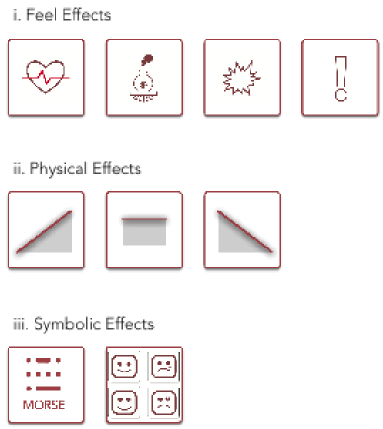


Figure 7.6: Graphical representation of haptic vocabularies and icons.

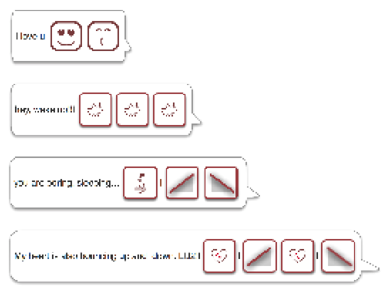


Figure 7.7: Some examples of expressive haptic messages embedded with normal text messages.

Spacer: keeps steady intensity; parameters are intensity and duration. This can be used for putting a delay (or spaces) between two haptic effects.

These effects create new haptic effects and can also be combined with feel effects. Such as the message Ramp-up — Motor followed by Ramp-down creates a new pattern that gradually increases the rumbling and then decays linearly as shown in Fig 5.

Type 3: Coded Effects

This type demonstrates symbolic vocabularies, such as one adapted from International Morse Code that consists of pre-stored pulses of dots and dashes. Other examples can be vibratese language [16], emoticon, and input from peripheral sensors.

7.2.4 Implementation and Preliminary Study

As a preliminary mockup, we developed an Android application on a Samsung S5 smartphone running Android 4.4.2. The Android API allows ON/OFF control of the embedded VT motor. A rough relationship between the duty cycle and perceived intensity was determined to create effects.

In a pilot, authors set parameters of three FEs defined by phrases ?racing pulse?, ?calm pulse? and ?high urgency?. These set parametric spaces for FEs defined by families pulse and urgency. From these families, new FEs were generated using the same logic as would in the normal language. For example, parameters for phrase ?hey, call me quick!!? are set same as that of high urgency (synonym) and parameters for phrase ?normal heartbeat? are inferred in-between those for calm and racing heartbeats.

We asked thirteen participants (8 males, 19-52 years old) to rate the haptic effects and associated phrase generated in the pilot on a scale from 0-5 (0: No agreement, 1: strong disagreement, 2: disagreement, 3: neutral, 4: agreement, 5: strong agreement). Participants could feel the haptic message as long as they wanted. The average (and standard deviation) ratings for calm, racing and urgency were 4.2 (0.7), 3.6 (1.6) and 4.2 (1.0) out of the scale of 5. We then asked the same participants to go through a mockup conversation that had five messages with both text and haptics. Participants played the haptics by tapping on the messages. Afterwards, they were asked to rate ?if available, will you use this application? and ?Does it communicate expressions?. Participants rated from ?yes? (1), ?not sure? (0), and ?no? (-1). The average (std. dev.) rating of usefulness was 0.77 (0.6) and that for expressiveness was 0.62 (0.5).

7.2.5 Concluding Remarks

In this early exploration, we implement and evaluate the use of haptic feedback in a messenger application on a typical mobile device. The application allows users to send and receive text and haptic messages, and to personalize haptic effects. The overall user ratings were positive, however a comprehensive study to analyze the effectiveness of feel effects for social networking is left for future. Our previous work has shown that feel effects improve users vocabulary and associate them to events in stories for adults and kids [9, 18]. Sharing personalized haptic messages in social interactions will be stimulating and exciting addition to the current networking tools.

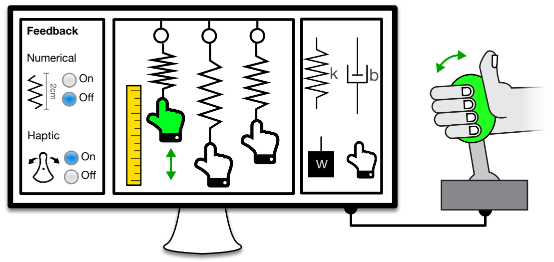


Figure 7.8: Students, teachers, and researchers can explore science, technology, engineering, and math (STEM) abstractions through low-fidelity haptics, incorporating elements into system designs.

7.3 RoughSketch

todo

7.4 HandsOn

Recognition of the value of a hands-on, embodied approach to learning dates to 1907, when Maria Montessori opened a school where she used *manipulatives* to teach a wide array of concepts ranging from mathematics to reading, e.g., by introducing the alphabet through children tracing their finger along large, cut-out letters [**?**]. Constructivist learning theories posit that well-designed manipulatives can assist understanding by grounding abstract concepts in concrete representations [58**?**], and are an accepted core principle in early math and science education, confirmed empirically [**?**].

More recently, digital technologies are radically altering learning environments. Massive Open Online Courses (MOOCs) expand access, games motivate, and with graphical simulations (e.g., PhET [**?**]), students can interact with abstractions to develop their understanding. However, these experiences are disembodied. Indirect contact via keyboard, mouse and screen introduces a barrier of abstraction that undermines the connection and path to understanding.

Haptic (touch-based) technology should bring benefits of physicality and embodied learning [**?**] to interactive virtual environments. It adds a sensory channel as another route to understanding [**?**]; when deployed appropriately, active exploration can improve understanding [**?**] and memory [**?**] of new concepts. Haptic tools have already shown promising results in many specializations, demographics and age groups, both to enhance lesson fidelity and to increase engagement and motivation through tangibility and interactivity; e.g., with devices like Geomagic

Touch[1] [**?**] and SPIDAR-G [**?**].

Unfortunately, existing approaches have both hardware and software limitations. Actuated learning tools introduce physical issues of cost, storage, and breakage; devices are too bulky, complex, or expensive for schools or self-learners. For software, it is hard for users to construct and explore their own haptic environments. Typically, users load a virtual system to interact with it haptically. This sidelines the rich learning potential of involving users with model construction [58]. We address hardware with the HapKit [**?**], a $50, simple, low-fidelity device constructed from 3d printed materials.

Our focus here is on software, with a new learning environment that lets users both construct and explore haptic systems. Until now, the only way for a user to construct a haptic system was by programming it herself. Our approach, inspired by Logo [58] and Scratch [**?**], is to ultimately provide much of the power of a programming language while hiding distracting complexity.

Approach and Present Objectives:

To study *how* to unlock the potential of hapticized virtual environments in STEM education, we need a viable front-end. To this end, we first established a *conceptual model* (*HandsOn*): central interface concepts, supported operations and language [**?**] that can be employed in a broad range of lessons involving physical exploration and design.

Next, we implemented the *HandsOn* conceptual model (CM) in *SpringSim*, a first-generation learning interface prototype narrowly focused in a module on mechanical springs and targeted at high school physics students. To render forces we used the HapKit, a simple device with a 3D-printable handle providing affordable, self-assembled 1 DOF force-feedback for about $50 USD. As an evaluation instrument, this single-lesson implementation allows us to (a) measure a given hardware platform's fidelity for a representative perceptual task; (b) attain insight into the kinds of lessons such a system can leverage; and (c) assess its learning-outcome efficacy relative to conventional methods. With these answers, we will be able to design a more powerful tool.

We report results from two user studies: (1) the HapKit's ability to display differentiable springs with and without graphical reinforcement, and (2) a qualitative evaluation of *SpringSim* for a carefully designed set of educational tasks. We confirm that the *SpringSim* interface and its conceptual model *HandsOn* are understandable and usable, describe the role of haptics compared to mouse input, and provide recommendations for future evaluation, lesson and tool design.

[1]Prev. Sensable Phantom www.geomagic.com/en/products/phantom-omni/overview

7.4.1 Tool Development: Conceptual Model and Interface

Our goal was to find a software model to use and evaluate low-cost force feedback in an educational setting. We began by choosing a device, establishing requirements, and exploring capabilities through use cases and prototypes. From this, we defined *HandsOn*. We then implemented essential features in a medium-fidelity prototype, *SpringSim*, for our user studies.

Initial design (requirements):

We established six guiding requirements. First, we developed initial prototypes with HapKit 2.0 through two pilot studies with middle school students (described in [**?**]). These highlighted two aspects of a practical, accessible approach for junior students: 1) no programming; instead 2) a graphical implementation of an exploratory interface within a lesson plan. We also needed to build on known benefits of traditional classroom practices, and enable learning-outcome comparison. We must 3) support the same *types* of traditional education tasks, e.g., let students compare and assemble spring networks as easily as in a hands-on physics lab; but also 4) *extend* them, to leverage the flexibility offered by a manipulative that is also virtual. Similarly, to support future formal comparisons, our model needs to 5) support both haptic and non-haptic (mouse) inputs. Finally, to ensure generality we also needed to 6) support diverse STEM topics, like physics, biology, and mathematics. Further design yielded a model that addressed these requirements: *HandsOn*.

Conceptual Model:

HandsOn is a programming-free (R1) graphical interface supporting learner exploration (R2), with a number of key *concepts*: *Interactive Playground*, *Hands*, *Design Palette*, *Objects*, *Properties*, *Haptic* and *Visual Controls*. Exploration is supported at various levels (Figure 7.9).

The *Interactive Playground* provides a virtual sandbox where users can interact with virtual environments (VE). *Hands* allow users to select, move, and manipulate components in the Interactive Playground. Control occurs with either the mouse or a haptic device to receive force-feedback (Figure 7.9A) (R5). In the design and modification phase, users can add or remove *objects* like springs, masses, gears, or electrons by dragging them to and from a *Design Palette* (R3). Once added to the scene, users can modify their physical properties (e.g., a spring constant k) and make changes to the VE (Figure 7.9B). After construction, the user can customize their interaction with their VE by adjusting *Visual Controls* and *Haptic Controls* options that extend interactions in new ways afforded by haptics (R4)

(Figure 7.9C). Because of the flexibility afforded by having multiple *objects* in the playground with multiple *Hands* for interaction points, and customization of interaction and feedback, *HandsOn* can support different STEM topics (R6), from biology to mathematics. To confirm the viability of this approach, we built an initial prototype with essential features: *SpringSim*.

Implemented Prototype:

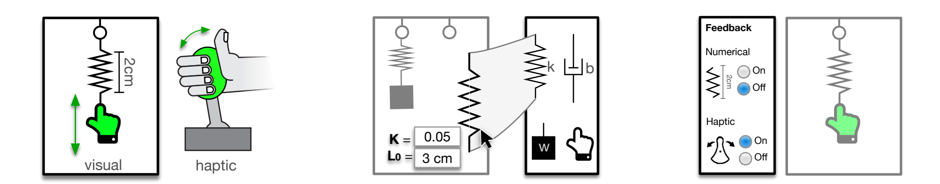
Our first *HandsOn* interface is *SpringSim* (Figure 7.10), which supports a spring lesson – spring systems are natural as a virtual environment of easily-controlled complexity. In *SpringSim*, *objects* include single springs and parallel spring systems, with properties spring rest length (cm), stiffness (N/m) and label. The *Design Palette* includes the *Spring Properties* and *Spring Generator* UI components. Implemented *Visual Controls* are toggling numerical displays of spring stiffness and force; *Haptic Controls* toggle HapKit feedback and output amplification. The open-source repository for SpringSim is available at https://github.com/gminaker/SpringSim.

7.4.2 Study 1: Perceptual Transparency

Before evaluating *SpringSim*, we needed to confirm that the HapKit could render spring values sufficiently for our qualitative analysis.

Methods:

14 non-STEM undergraduate students (8 females) participated in a two-alternative, forced choice test with two counterbalanced within-subject conditions: *HapKit + Dynamic Graphics*, and *HapKit + Static Graphics* (Figure 7.11). Three spring pairs (15/35, 35/55 and 55/75 N/m) were each presented five times per condition, in random order. For each pair, participants indicated which spring felt more stiff,



A) Interact with the system in the *Interactive Playground* using a selected *Hand*, manipulating and monitoring state via multimodal feedback.

B) Create and Modify the system with a *Design Palette*, adding or removing *Objects* and changing object properties.

C) Customize interaction itself for learners, teachers, and researchers, adjusting input/output modalities with *Visual Controls* and *Haptic Controls*.

Figure 7.9: The *HandsOn* CM enables three kinds of exploration based on requirements.

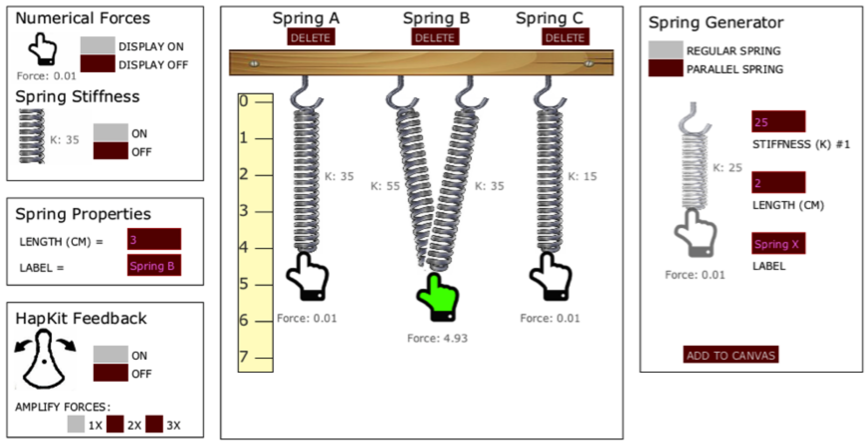


Figure 7.10: *SpringSim* interface, a *HandsOn* sandbox for a single lesson module on springs.

and rated task difficulty on a 20-point scale. Following each condition, participants rated overall condition difficulty, mental demand, effort, and frustration on 20-point scales derived from the NASA TLX [**?**]. Following the completion of both conditions, a semi-structured interview was conducted to address any critical incidents. Each session lasted 20-30 minutes.

Results:

All tests used a 5% level of significance and passed test assumptions.

Accuracy: A logistical regression model was trained on task accuracy with spring-pair and condition as factors. No interaction was detected; spring-pair was the only significant factor. Post-hoc analysis revealed that spring-pair #1 (15/35 N/m) was significantly less accurate than spring-pair #2 (35/55; p=0.0467). Performance averaged 88.57% (15/35), 96.49% (35/55), and 94.45% (55/75).

Time: Task time ranged from 3-160s (median 117s, mean 96.41s, sd 47.57s). In a 3-way ANOVA (participant, spring-pair, and visualization condition) only participant was significant ($F(13,336) = 4.17$ $p = 1.947e-06$).

Difficulty rating: A 3-way ANOVA (factors: participant, spring-pair, and visualization condition) detected one two-way interaction between participant and spring pair ($F(26,336) = 2.10$, $p = 0.00165$).

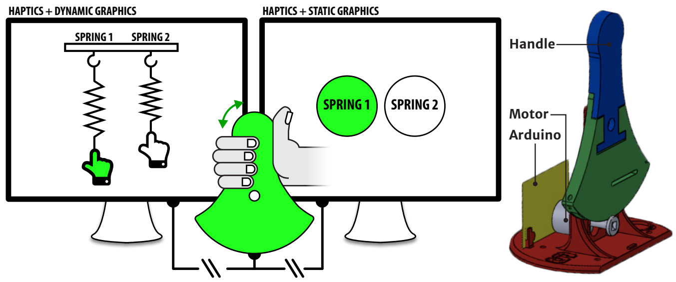


Figure 7.11: In the *Hapkit+Dynamic Graphics* condition, graphical springs responded to input (left); static images were rendered in the *Hapkit+Static Graphics* condition (right); in both, HapKit 3.0[**?**] was used as an input/force-feedback device (far right).

Discussion:

Study 1 revealed that (a) for stiffness intervals 15/35/55/75 N/m, the HapKit provides distinguishability equivalent to dynamic graphics. Individual differences influenced difficulty and speed, suggesting that learning interfaces may need to accommodate this variability. (b) Accuracy was not dependent on individual differences, suggesting that learning interfaces can consider task time and perceived difficulty separately from accuracy when using the HapKit (at least, for these force ranges). (c) Performance was mostly above 90%, and confidence intervals for our small sample size estimate no lower than 82% accuracy at the lowest (15/35). We speculate that the HapKit's natural dynamics are more pronounced at lower rendered forces, and may interfere with perceptibility.

7.4.3 Study 2: Tool Usability and Educational Insights

Methods:

10 non-STEM participants (1st and 2nd year university undergrads with up to first year physics training, 6 female, 17-20 years) volunteered for 45-60 minute sessions. After an introductory survey, participants were randomly assigned to one of two conditions, *Mouse* (4 participants, M1-4) or *Hapkit* (H1-6). HapKit 3.0 was calibrated for force consistency between participants. After allowing participants to freely explore *SpringSim*, a survey assessed understanding and usability of various *SpringSim* interface components; misunderstood components were clarified. Three exit surveys elicited value of *SpringSim* components on 7-point Likert scales, cognitive load [**?**], understanding, and curiosity on 20-point scales, and preferred

Task	Bloom	Description
1	Understand (2)	Rank three springs in order from least to most stiff
2	Understand (2)	Plot the relationship between displacement and force for two springs.
3	Apply (3)	Estimate the stiffness of an unknown spring, given two reference springs with known stiffness value
4	Analyze (4)	Predict the behaviour of springs in parallel.
5	Create (6)	Design a parallel spring system that uses two springs to behave like an individual spring of stiffness 55 N/m.
6	Apply (3)	Predict the behaviour of springs in series.
7	Evaluate (5)	Describe any relationships you have noticed between spring force, displacement, and stiffness.

Table 7.1: Learning tasks used with *SpringSim* in Study 2. *Bloom* level is a measure of learning goal sophistication [**?**]

learning modality [**?**], respectively.

Learning Tasks:

We iteratively designed and piloted a task battery of escalating learning-goal sophistication [**?**] to expose strategies for force feedback use and general problem-solving (Table 7.1). Tasks did not require physics knowledge, and were suitable for both mouse and HapKit input.

Analysis:

We conducted t-tests on self-reported understanding, cognitive load, engagement, understanding, curiosity; and on objective metrics of time-on-task and number of spring interactions. Qualitative analysis of video and interview data used grounded theory methods of memoing and open & closed coding [11]. Together, these yielded insight into the usability of *SpringSim* and the *HandsOn* CM, and several themes describing the role of haptics in our tasks. Two participants were excluded from analysis of Task 1 due to technical failure.

Results - Usability:

After free exploration of *SpringSim*, participants rated their understanding of CM objects (yes/no) and their ease-of-use [1-7]: *Ruler* (10/10, 7.0), *Numerical Force Display* (10/10, 6.5), *Playground* (10/10, 6.0), *Hand* (9/10, 6.0), *Spring Properties* (9/10, 6.0), *Spring Generator* (**7/10, 5.0**), *HapKit* (6/6, **4.5**), and *Haptic Feedback Controls* (5/6, **4.5**). While generally usability was good, interface clarity needed improvement in highlighted cases. Participants specifically noted confusion on radio button affordances, and *Spring Generator* input fields (due to redundant availability in *Spring properties*).

Results - Task Suitability for Haptic Research:

Regardless of prior physics knowledge, all participants were able to complete education tasks 1-6 (Table 7.1) in the allotted 60 minutes. We found no evidence that any task favoured one condition over another. When participants in the mouse condition were asked how their workflow would change with physical springs, participants weren't sure: "I don't know if that would've given me more information" (M4).

Results - Haptics & Learning Strategies:

We observed several themes relating to the influence of force feedback on a student's learning strategy.

Haptics creates new, dominating strategies. Learning strategies used by participants in the HapKit condition (H1-6) were more diverse than those in the mouse condition (M1-4). In Task 1, M1-4 all followed the same strategy, displacing all 3 springs the same distance and comparing the numerical force required to displace them. They then correctly inferred that higher forces are associated with stiffer springs (the *displace-and-compare* strategy).

By contrast, all 5 H participants included in analyses (H2 excluded due to technical failure) used force-feedback as part of their approach to Task 1. H1 describes applying the same force to the HapKit across all 3 springs, recording displacement to solve the task, while H5 described looking at the speed at which the HapKit was able to move back-and-forth in making his determination of stiffness, rather than through direct force-feedback of the device. Only H6 indicated that he "looked at the numbers for a sec", but no participant fully used the *displace-and-compare* strategy we observed for M participants.

While the single-strategy approach worked for easy tasks, it was linked to errors and dead-ends in at least one instance in the mouse condition. In Task 5, M2-4 used *displace-and-compare* to validate their newly designed spring; M1 did not seek verification of his design. In contrast, H1,2,5,6 used haptic feedback to verify their designs. They did this by comparing how stiff their parallel spring system felt to a target reference spring. H4 guessed at an answer without verification. H3 used the *displace-and-compare* strategy, checking that equal forces were required for equal displacement.

Haptic impressions of springs are enduring and transferrable. HapKit participants were able to use their previous explorations to solve problems. In Task 3, M1-4 interacted with all three springs to find a ratio between force and stiffness. However, H participants interacted with springs fewer times (mean 1.5, sd 3.21) than M (6, sd 1) (p=0.018). H2-4,6 did not interact with any springs, and H1 interacted with only one. This was because they had already interacted with the springs in previous questions: "I remember spring C was less stiff" (H3). Further suggesting the strength of haptic impressions, when H1 designed an inaccurate spring system for Task 5 (k=80N/m vs. expected k=55N/m), she described the haptics as overriding the visual feedback: "they just felt similar. Even though the numbers weren't really relating to what I thought." Similarly, H2 arrived at an approximate result (k=40N/m), after using force-feedback and acknowledges "... [it's] slightly less than the reference spring, but it's closer."

Haptics associated with increases in self-reported curiosity and understanding. Participants' self-reported curiosity significantly increased over the course of HapKit sessions from a mean of 6.3 (sd 3.83) to 10.8 (sd 3.92) in the Hapkit condition (p=0.041). No significant changes in curiosity were detected in the mouse condition. Participants' self-reported understanding significantly increased over the course of HapKit sessions from a mean of 3.67 (sd 4.03) to 11.83 (sd 3.19) (p=0.014). No significant changes in understanding were detected in the mouse condition (before: 9.25, sd 5.32; after: 9.25, sd 5.32; p=0.77).

In interviews, participants commonly made references to how the HapKit influenced their understanding: "I can use this thing for help if I really need some physical, real-world stimuli" (H5); "almost all of my thinking was based on how the spring [HapKit] ended up reacting to it" (H6). M2, who had a stronger physics background than others (IB Physics), was the only user to report a drop in curiosity and understanding over the course of the physics tasks, despite initial excitement: "the fun part is messing around with [SpringSim]," he exclaimed near the beginning of the exploratory phase.

7.4.4 Study 2 Discussion

Tool and Tasks: Suitability for Learning and as Study Platform

Adequacy and comprehensibility of underlying model: Overall, *HandsOn* concepts proved an effective and comprehensible skeleton for *SpringSim*. Specific implementations rather than concepts themselves appeared to be the source of the reported confusions, and we observed that *HandsOn* should be extended with additional measurement tools (e.g., protractors, scales, calculators, etc).

SpringSim performance: This *SpringSim* implementation adequately supported most students in finishing learning tasks; extending available objects, properties and tasks will support advanced students as well. Future iterations should more clearly map *Design Palette* elements to the objects they support, increasing rendering fidelity and reconsider colors to avoid straightforward affordance issues. While participants did not heavily use haptic and visual controls, we anticipate these will be important for instructor and researcher use.

Learning task suitability: The learning tasks used here were fairly robust to time constraints of user-study conditions, did not require previous physics knowledge, avoided bias from standardized physics lessons, and exposed haptics utilization strategies without penalizing non-haptic controls. Currently, the task set ends by asking students to predict a serial system's behavior; some students found predicting new configurations a large jump. Future task-set iterations could support integrative, prediction-type questions with interface elements that are successively exposed to allow prediction testing.

Evidence of the Role of Force Feedback in Learning

Curiosity and understanding leading to exploration: Self-reported curiosity and understanding increased when forces were present. While these trends must be verified, curiosity is of interest since it can lead to more meaningful and self-driven interactions. Iterations on both tasks and tool should support this urge with an interface and framing that supports curiosity-driven exploration.

Alternative strategies enabled by force feedback: The HapKit's additional feedback modality enabled alternative task workflows, e.g., estimations of force appeared to supplant mathematical strategies for stiffness estimation. While possibly risky as a crutch, force assessments might be a useful step for students not ready for technical approaches (e.g., M3/Task 3 when stalled in attempting cross-multiplication). Future task-set iterations could encourage more *balanced* strategy use, e.g. mathematical *and* perceptual rather than primarily perceptual.

HapKit salience, resolution & implications: Overall, HapKit 3.0's fidelity was enough to assist participants verify a correct hypothesis. However, those who started with an *incorrect* hypothesis and used only HapKit to test it generally arrived at solutions that improved but were still inaccurate. Given the confidence that forces instilled, this is an important consideration. A formal device characterization will allow us to keep tasks within viable limits; we can also consider using low-fidelity forces more for reinforcement and exploratory scenarios.

Limitations and Next Steps:

Our studies were small and used non-STEM university students as a proxy for high-school learners. Despite both limitations, they were useful for our current needs (rich, initial feedback establishing suitability and usability for *HandsOn* through *SpringSim*); but may overestimate general academic ability and maturity. As we we move into evaluation of learning outcome impact, larger and more targeted studies are imperative.

Future interfaces can both increase physical model complexity and breadth (e.g., complex mass-spring-damper systems), and extend *HandsOn* for more abstract education topics, such as trigonometry. We also plan to extend the *Playground* to support more engaging, open-ended student design challenges, such as obstacle courses using trigonometry concepts; this in turn requires new measurement tools and tasks that are more exploratory and open-ended.

7.4.5 Conclusions

Haptic feedback's potential in STEM education use can only be accessed with a comprehensible, extendable, and transparent front-end. We present *HandsOn*, a conceptual skeleton for interfaces incorporating virtual forces into learning tasks, and assess its first implementation, *SpringSim* and task set. Our findings (on interface usability, task effectiveness, and impact of haptic feedback on learning strategies, understanding and curiosity) underscore this approach's promise, as we proceed to study haptic influence on learning outcomes themselves.

7.5 CuddleBit

Chapter 8

Ground: Lessons from Hapticians

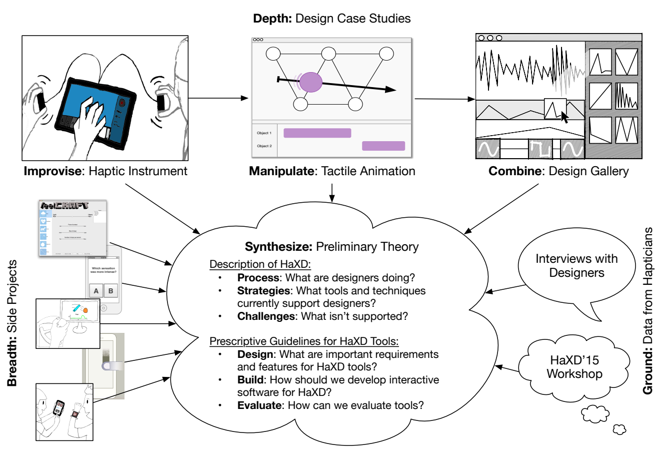


Figure 8.1: Planned synthesis of data for a preliminary theory of haptic experience design.

The three case studies provide rich but focused data on how to create vibrotactile (VT) experience design tools. To complement these studies, I propose to gather information more broadly to generalize the haptic design process to other use cases and ground in haptician experience (Figure 8.1). In this way, my final contributions will draw from several sources: 1) the three in-depth design studies, 2) insight gathered from several side projects, and 3) two sets of grounded data: interviews with professional designers, and a workshop at World Haptics '15.

Here, I describe the data collection process, then illustrate possible applications and forms for these contributions. However, any resulting theory will be emergent from the data, and can take many forms. To accomplish this in a principled way, I plan to use *memoing* and *constant comparison* [11], looking for common threads between data and double-checking conclusions as new theoretical developments appear. This theory will also draw from a literature review of design theory, summarized in this document's related work section.

8.1 Data Collection

In this section, I list the different sources I intend to use to collect data for theory design.

8.1.1 Vibrotactile design case studies

Each of the design case studies previously described investigates a specific user group working on a specific VT device with a specific software tool. Through my experience of gathering requirements, creating tools (through design and development), and evaluating them, I will have first-hand knowledge of supporting aspects of VT sensation design. Each also produces a small vignette of haptic design in action, giving us glimpses of the design process.

8.1.2 Side projects

In addition to the three design case studies that form this proposal, several side-projects are planned or underway as collaborative efforts. In these side-projects, I take an organizational or supervisory role, often as summer projects conducted by undergraduates.

FeelCraft and Feel Messenger are collaborations with Disney Research members Ali Israr and Siyan Zhao, looking at distributing and customizing haptic effects in a consumer setting with low-fidelity rumble motor devices. I take a haptic designer role to gain a personal understanding of the process, and a software engineer role to understand relevant architectures.

CyberHap is a collaboration between UBC and Stanford looking at force-feedback devices in education; a large team is involved with undergraduate Gordon Minaker leading development of a teaching interface since February 2015, co-supervised by PI Dr. Karon MacLean and me.

CuddleBit is a project inspired by the Haptic Creature and CuddleBot project. A small, breathing and vibrating robot will be designed along with a behaviour prototyping tool in summer 2015. I supervise undergraduate Paul Bucci in this project exploring multiple modalities and potential for receiving input through a sensor.

HapTurk is a collaboration with PhD candidate Hasti Seifi on different techniques to crowdsource feedback on VT icons. Master's student Salma Kashani and undergraduate Matthew Chun are developing visualizations and low-fidelity VT icons during summer 2015.

RoughSketch is a painting application for the TPad Phone, a variable-friction mobile device, for the World Haptics 2015 Student Innovation Challenge. Undergraduates Brenna Li, Paul Bucci, and Gordon Minaker are all fellow team members. Variable friction is a significant contrast to VT sensations as it is intrinsically connected to input: no sensation can be felt without active movement by the user.

8.1.3 Grounded Data

A corpus of interviews with professional haptic designers has already been collected by UBC alumni Colin Swindells during his PhD, but has never been published. I will analyze these interviews to further ground our findings with real-world haptic designers.

To complement this we turn to the research community who do design as part of their work. The planned Workshop on Haptic Experience Design (http://oliverschneider.ca/HaXD) at World Haptics 2015 will also provide a data source. At this workshop, 4 experts of haptic experience design will speak, participants will reveal their own design challenges in a brainstorming activity, and the ensuing panel discussion should help illuminate practices and paths for future work.

8.2 Possible Format

While the theory can take many forms, I hope to characterize haptic experience design and contrast it with other design fields, especially graphic and audio design. I hope to find both descriptive and prescriptive results, including current practices, an identification of challenges uniquely facing haptic designers, and guidelines for designers and developers of haptic design support tools.

8.2.1 Descriptive Contributions: HaXD Process as Requirements

My first goal is to describe the **processes** employed by haptic designers. This could manifest, for example, as a catalogue of existing haptic design tools, appropriated tools (e.g., using a sound editor to create VT icons), techniques (e.g., design philosophies like Haptic Sketching), resources (e.g., libraries and APIs), platforms (what devices designers are using), practices in haptics education (undergraduate or graduate level courses), and tasks undertaken by haptic designers.

For example, to share of haptic experiences, haptic designers create demos to spread awareness of haptic research and gain feedback from peers. This is so ingrained into the culture of haptic research that recently a demo-only conference was launched: Asia Haptics.

Once collecting a description of current practices, I expect we might be able to identify **challenges** and **strategies** to addressing those strategies, including the ecosystem of available tools: what is working well, and what is broken. Using our example of collaboration and demos, we might see that in-person demos are effective, but remote collaboration or asynchronous sharing is challenging. Available tools include videos and visualizations of demos to explain concepts in lieu of the demo itself.

8.2.2 Prescriptive Contributions: Guidelines for HaXD Software Tools

After describing HaXD as a set of requirements, I will then develop guidelines for how to built supportive interactive software tools. Right now, I plan to organize this into three aspects: how to **design** tools, including important features relevant to different stages of HaXD; how to **build** tools, including relevant software architectures and ways to address technical challenges; and how to **evaluate** tools, methods to capture designer experience and inform future design.

Hypothetical use cases might best explain this contribution. One examples is using these guidelines for knowledge transfer to industry. I could use these guidelines to advise or create design software for companies developing haptic hardware platforms (such as the TPad team and UltraHaptics) or software platforms (such as Immersion and Phidgets), bridging the gap from research to industry application. Another example could be dissemination through haptics education. Developing a module for a haptic design course, such as CPSC 543, is an accessible way to encapsulate and test these ideas. This could also manifest in a multi-day workshop, similar to Camille Moussette's Haptic Sketching workshops, to validate ideas at different institutions.

8.3 Deliverables and Risk

There are two expected deliverables from this theory development. First, the HaXD'15 workshop on Haptic Experience design is planned, piloted, and scheduled for World Haptics in June, 2015. To get the most out of this workshop, photographs and notes will be recorded. Afterwards, a very small digest piece debriefing the workshop is planned in winter 2016; this may be submitted on its own as an short paper if an appropriate venue is available (e.g., a special journal issue similar to [38]), or subsumed into the second deliverable.

The second deliverable is a retrospective piece on our findings from all the data sources found here, but with a focus on data from haptic designer interviews. This interview data has already been collected by UBC alumnus Colin Swindells. I plan to digest and analyze those interviews in winter 2016 to generate requirements grounded in designer experience. This will likely be combined with synthesized findings from the three design studies and several side projects. To mitigate risk, we can combine interview findings to a greater or lesser extent with other data sources. If the interviews have a great deal of information, they could be a valuable contribution on their own. If not, I expect them to supplement our other data sources. This document will likely be submitted as a full paper to a peer-reviewed conference or journal.

Within each project, we mitigate risk through strategic planning and study design; many of these projects do not have to be successful to provide input. For example, HapTurk may never actually be deployed, but could still articulate the challenge of a large-scale, remote haptic user study. In addition, risk is partially managed through sheer attrition: one or two side projects or data sources could provide limited feedback and we would still have a diverse set of information. However, I will note that initial investigation has already been useful.

Chapter 9

Conclusion

In this dissertation we demonstrated contributions and other contributions. Our contributions revealed important takeaways and highlighted a need for problem, but ultimately showed a success for goal. Future work will contribute further contributions on problem and related goals. Together, we will save the world through science.

Bibliography

[1] D. Alles. Information Transmission by Phantom Sensations. *IEEE Transactions on Man Machine Systems*, 11(1):85–91, Mar. 1970. ISSN 0536-1540. doi:10.1109/TMMS.1970.299967. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4081935. → pages 29, 33

[2] Android Open Source Project. Android Developers, 2012. URL http://developer.android.com/index.html. → pages 17

[3] S. Brewster and L. M. Brown. Tactons: structured tactile messages for non-visual information display. In *AUIC '04*, pages 15–23, Jan. 2004. URL http://dl.acm.org/citation.cfm?id=976310.976313. → pages 11, 17, 45, 61

[4] S. A. Brewster, P. C. Wright, and A. D. N. Edwards. An evaluation of earcons for use in auditory human-computer interfaces. In *CHI '93)*, pages 222–227, New York, USA, May 1993. ACM Press. ISBN 0897915755. doi:10.1145/169059.169179. URL http://dl.acm.org/citation.cfm?id=169059.169179. → pages 11

[5] L. Brown, S. Brewster, and H. Purchase. A First Investigation into the Effectiveness of Tactons. In *World Haptics '05*, pages 167–176. IEEE, 2005. ISBN 0-7695-2310-2. doi:10.1109/WHC.2005.6. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1406930. → pages 11

[6] L. M. Brown, S. A. Brewster, and H. C. Purchase. Tactile crescendos and sforzandos. In *CHI Extened Abstracts '06*, pages 610–615, New York, USA, Apr. 2006. ACM Press. ISBN 1595932984. doi:10.1145/1125451.1125578. URL http://dl.acm.org/citation.cfm?id=1125451.1125578. → pages 11, 17, 45

[7] L. M. Brown, S. A. Brewster, and H. C. Purchase. Multidimensional tactons for non-visual information presentation in mobile devices. In *MobileHCI*

'06, pages 231–238, New York, USA, Sept. 2006. ACM Press. ISBN 1595933905. doi:10.1145/1152215.1152265. URL http://dl.acm.org/citation.cfm?id=1152215.1152265. → pages 11, 17, 61, 63

[8] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007. → pages 12, 13, 43, 46

[9] J. Cha, Y.-S. Ho, Y. Kim, J. Ryu, and I. Oakley. A Framework for Haptic Broadcasting. *IEEE Multimedia*, 16(3):16–27, July 2009. ISSN 1070-986X. doi:10.1109/MMUL.2009.42. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5228689. → pages 11

[10] A. Chan, K. MacLean, and J. McGrenere. Designing haptic icons to support collaborative turn-taking. *International Journal of Human-Computer Studies*, 66(6):333–355, 2008. URL http://www.sciencedirect.com/science/article/pii/S1071581907001565. → pages 11, 63, 65, 67, 68

[11] J. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc., 3 edition, 2008. → pages 19, 36, 49, 83, 103, 110

[12] J. W. Creswell. *Qualitative inquiry and research design: choosing among five approaches*. Sage Publications Inc, 3rd edition, 2013. → pages 14, 20

[13] N. Cross. *Designerly Ways of Knowing*. Springer-Verlag London Limited, 2006. → pages 13

[14] D. Cuartielles, A. Goransson, T. Olsson, and S. Stenslie. Developing Visual Editors for High-Resolution Haptic Patterns. In *HAID'12 Posters and Demos*, pages 42–44, 2012. URL http://www.english.certec.lth.se/haptics/HAID12/-proceedings.pdf. → pages 12, 26

[15] H. Culbertson, J. Unwin, and K. J. Kuchenbecker. Modeling and rendering realistic textures from unconstrained tool-surface interactions. *IEEE Transactions on Haptics*, 7(3):381–93, Jan. 2014. ISSN 2329-4051. doi:10.1109/TOH.2014.2316797. URL http://www.ncbi.nlm.nih.gov/pubmed/25248220. → pages 9, 10, 43

[16] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, M. Christie, and A. Lecuyer. Toward Haptic Cinematography: Enhancing Movie Experiences with Camera-Based Haptic Effects. *IEEE MultiMedia*, 21(2):11–21, 2014. ISSN 1070-986X. doi:10.1109/MMUL.2013.64. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6695749. → pages 11

[17] S. Dow, J. Fortuna, D. Schwartz, B. Altringer, D. Schwartz, and S. Klemmer. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *CHI '11*, pages 2807–2816. ACM Press, May 2011. ISBN 9781450302289. doi:10.1145/1978942.1979359. URL http://dl.acm.org/citation.cfm?id=1978942.1979359. → pages 13, 43, 56

[18] Z. Eitan and I. Rothschild. How music touches: Musical parameters and listeners' audio-tactile metaphorical mappings. *Psychology of Music*, 39(4): 449–467, Nov. 2010. ISSN 0305-7356. doi:10.1177/0305735610377592. URL http://pom.sagepub.com/content/39/4/449.short. → pages 11

[19] C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1):39–58, Jan. 1991. ISSN 00010782. doi:10.1145/99977.99987. URL http://dl.acm.org/citation.cfm?id=99977.99987. → pages 13

[20] M. Enriquez and K. MacLean. The hapticon editor: a tool in support of haptic communication research. In *HAPTICS '03*, pages 356–362. IEEE Comput. Soc, 2003. ISBN 0-7695-1890-7. doi:10.1109/HAPTIC.2003.1191310. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1191310. → pages 11, 23, 26, 29, 45, 46

[21] M. Enriquez, K. MacLean, and C. Chita. Haptic phonemes: basic building blocks of haptic communication. In *ICMI '06*, page 302, New York, USA, Nov. 2006. ACM Press. ISBN 159593541X. doi:10.1145/1180995.1181053. URL http://dl.acm.org/citation.cfm?id=1180995.1181053. → pages 11, 45, 46

[22] FMOD. FMOD Ex API, 2013. URL http://www.fmod.org. → pages 17

[23] J. Forsslund, M. Yip, and E.-L. Sallnäs. WoodenHaptics. In *TEI '14*, pages 133–140. ACM Press, Jan. 2015. ISBN 9781450333054. doi:10.1145/2677199.2680595. URL http://dl.acm.org/citation.cfm?id=2677199.2680595. → pages 10

[24] S. Greenberg and C. Fitchett. Phidgets: Easy development of physical interfaces through physical widgets. In *UIST '01*, page 209, New York, New York, USA, Nov. 2001. ACM Press. ISBN 158113438X. doi:10.1145/502348.502388. URL http://dl.acm.org/citation.cfm?id=502348.502388. → pages 10

[25] S. Guest, J. M. Dessirier, A. Mehrabyan, F. McGlone, G. Essick, G. Gescheider, A. Fontana, R. Xiong, R. Ackerley, and K. Blot. The development and validation of sensory and emotional scales of touch perception. *Attention, perception & psychophysics*, 73(2):531–50, Feb. 2011. ISSN 1943-393X. doi:10.3758/s13414-010-0037-y. URL http://www.ncbi.nlm.nih.gov/pubmed/21264727. → pages 63, 64

[26] E. Gunther, G. Davenport, and S. O'Modhrain. Cutaneous grooves: composing for the sense of touch. In *NIME '02*, pages 1–6, New York, NY, USA, May 2002. ISBN 1-87465365-8. URL http://dl.acm.org/citation.cfm?id=1085171.1085181. → pages 11, 17, 45

[27] B. Hartmann, L. Yu, A. Allison, Y. Yang, and S. R. Klemmer. Design as Exploration : Creating Interface Alternatives through Parallel Authoring and Runtime Tuning. In *UIST*, pages 91–100, 2008. ISBN 9781595939753. → pages 13

[28] V. Hayward. A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store. *Brain research bulletin*, 75(6):742–52, Apr. 2008. ISSN 0361-9230. doi:10.1016/j.brainresbull.2008.01.008. URL http://www.sciencedirect.com/science/article/pii/S0361923008000178. → pages 10, 26, 63

[29] V. Hayward and K. Maclean. Do it yourself haptics: part I. *IEEE Robotics & Automation Magazine*, 14(4):88–104, Dec. 2007. ISSN 1070-9932. doi:10.1109/M-RA.2007.907921. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4437756. → pages 10

[30] S. R. Herring, C.-C. Chang, J. Krantzler, and B. P. Bailey. Getting inspired! Understanding How and Why Examples are Used in Creative Design Practice. In *CHI '09*, pages 87–96, New York, New York, USA, Apr. 2009. ACM Press. ISBN 9781605582467. doi:10.1145/1518701.1518717. URL http://dl.acm.org/citation.cfm?id=1518701.1518717. → pages 12, 13, 26, 43, 46

[31] K. Hong, J. Lee, and S. Choi. Demonstration-based vibrotactile pattern authoring. In *TEI '13*, page 219, New York, New York, USA, Feb. 2013. ACM Press. ISBN 9781450318983. doi:10.1145/2460625.2460660. URL http://dl.acm.org/citation.cfm?id=2460625.2460660. → pages 12

[32] E. Hutchins. *Cognition in the Wild*, volume 19. MIT Press, 1995. ISBN 0262082314. doi:10.1098/rsbl.2011.0352. URL http://www.ida.liu.se/~nilda/CST-papers/Hutchins.pdf. → pages 11, 13

[33] I. Hwang, J. Seo, M. Kim, and S. Choi. Vibrotactile Perceived Intensity for Mobile Devices as a Function of Direction, Amplitude, and Frequency. *IEEE Transactions on Haptics*, 6(3):352–362, July 2013. ISSN 1939-1412. doi:10.1109/TOH.2013.2. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6420831. → pages 61

[34] A. Israr and I. Poupyrev. Tactile brush: drawing on skin with a tactile grid display. In *CHI '11*, pages 2019–2028, Vancouver, BC, May 2011. ACM Press. ISBN 9781450302289. doi:10.1145/1978942.1979235. URL http://dl.acm.org/citation.cfm?id=1978942.1979235. → pages 27, 29, 30, 32, 33, 45, 57, 63

[35] A. Israr, S. Zhao, K. Schwalje, R. Klatzky, and J. Lehman. Feel Effects: Enriching Storytelling with Haptic Feedback. *Transactions on Applied Perception (TAP)*, 11(3), 2014. → pages 10, 11, 43, 44, 46, 61, 63

[36] A. Israr, S. Zhao, and O. Schneider. Exploring Embedded Haptics for Social Networking and Interactions. In *CHI Extended Abstracts '15*, pages 1899–1904, New York, New York, USA, Apr. 2015. ACM Press. ISBN 9781450331463. doi:10.1145/2702613.2732814. URL http://dl.acm.org/citation.cfm?id=2702613.2732814. → pages 65, 68, 69

[37] C. V. Jansson-Boyd. Touch matters: exploring the relationship between consumption and tactile interaction. *Social Semiotics*, 21(4):531–546, Sept. 2011. ISSN 1035-0330. doi:10.1080/10350330.2011.591996. URL http://dx.doi.org/10.1080/10350330.2011.591996. → pages 11, 23, 61, 64

[38] L. Jones. News from the Field: Courses in Haptics. *Transactions on Haptics*, 7(4):413–414, 2014. → pages 11, 113

[39] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000. → pages 9

[40] Y. Kim, J. Cha, I. Oakley, and J. Ryu. Exploring Tactile Movies: An Initial Tactile Glove Design and Concept Evaluation. *IEEE Multimedia*, PP(99):1, 2009. ISSN 1070-986X. doi:10.1109/MMUL.2009.63. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5255212. → pages 11, 12, 26, 27, 65

[41] D. Ledo, M. A. Nacenta, N. Marquardt, S. Boring, and S. Greenberg. The HapticTouch toolkit. In *TEI '12*, pages 115–122, New York, USA, Feb. 2012. ACM Press. ISBN 9781450311748. doi:10.1145/2148131.2148157. URL http://dl.acm.org/citation.cfm?id=2148131.2148157. → pages 10

[42] B. Lee, S. Srivastava, R. Kumar, R. Brafman, and S. R. Klemmer. Designing with interactive example galleries. In *CHI '10*, pages 2257–2266, New York, New York, USA, Apr. 2010. ACM Press. ISBN 9781605589299. doi:10.1145/1753326.1753667. URL http://dl.acm.org/citation.cfm?id=1753326.1753667. → pages 43, 44, 46, 47, 56

[43] J. Lee and S. Choi. Evaluation of vibrotactile pattern design using vibrotactile score. In *HAPTICS '12*, pages 231–238. IEEE, Mar. 2012. ISBN 978-1-4673-0809-0. doi:10.1109/HAPTIC.2012.6183796. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6183796. → pages 11, 12, 23, 26

[44] J. Lee, J. Ryu, and S. Choi. Vibrotactile score: A score metaphor for designing vibrotactile patterns. In *World Haptics '09*, pages 302–307. IEEE, Mar. 2009. ISBN 978-1-4244-3858-7. doi:10.1109/WHC.2009.4810816. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4810816. → pages 11, 16, 23

[45] K. Maclean and M. Enriquez. Perceptual design of haptic icons. In *Eurohaptics*, pages 351–363, 2003. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.138.6172. → pages 11, 45

[46] K. MacLean and V. Hayward. Do It Yourself Haptics: Part II [Tutorial]. *IEEE Robotics & Automation Magazine*, 15(1):104–119, Mar. 2008. ISSN 1070-9932. doi:10.1109/M-RA.2007.914919. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4476335. → pages 10

[47] K. E. MacLean. Putting Haptics into the Ambience. *IEEE Transactions on Haptics*, 2(3):123–135, July 2009. ISSN 1939-1412.

doi:10.1109/TOH.2009.33. URL
http://www.computer.org/portal/web/csdl/doi?doc=doi/10.1109/TOH.2009.33http://portal.acm.org/citation.cfm?id=1608573.1608735. → pages 10

[48] J. Marks, W. Ruml, K. Ryall, J. Seims, S. Shieber, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, and H. Pfister. Design galleries. In *SIGGRAPH '97*, pages 389–400, New York, New York, USA, Aug. 1997. ACM Press. ISBN 0897918967. doi:10.1145/258734.258887. URL http://dl.acm.org/citation.cfm?id=258734.258887. → pages 43, 44, 46, 47

[49] S. O. Modhrain and I. Oakley. Touch TV : Adding Feeling to Broadcast Media. 2001. → pages 11, 65

[50] C. Moussette. Sketching in Hardware and Building Interaction Design: tools, toolkits and an attitude for Interaction Designers. In *Proc. of Design Research Society*, 2010. URL https://public.me.com/intuitive. → pages 11

[51] C. Moussette and R. Banks. Designing through making. In *TEI '11*, pages 279–282, New York, USA, Jan. 2011. ACM Press. ISBN 9781450304788. doi:10.1145/1935701.1935763. URL http://dl.acm.org/citation.cfm?id=1935701.1935763. → pages 11, 23, 26

[52] C. Moustakas. *Phenomenological Research Methods*. SAGE Publications, Inc., 1994. ISBN 978-0-8039-5798-5. → pages 14, 18, 36, 49

[53] S. Mueller, T. Mohr, K. Guenther, J. Frohnhofen, and P. Baudisch. faBrickation - Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks. In *CHI '14*, pages 3827–3834. ACM Press, Apr. 2014. ISBN 9781450324731. doi:10.1145/2556288.2557005. URL http://dl.acm.org/citation.cfm?id=2611222.2557005. → pages 10

[54] M. Obrist, S. A. Seah, and S. Subramanian. Talking about tactile experiences. In *CHI '13*, pages 1659–1668. ACM Press, 2013. ISBN 9781450318990. doi:10.1145/2470654.2466220. URL http://dl.acm.org/citation.cfm?doid=2470654.2466220. → pages 11, 14, 23, 45, 61, 63, 64

[55] S. Okamoto, H. Nagano, and Y. Yamada. Psychophysical Dimensions of Tactile Perception of Textures. *IEEE Transactions on Haptics*, 6(1):81–93, Jan. 2013. ISSN 1939-1412. doi:10.1109/TOH.2012.32. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6216375. → pages 11, 61

[56] A. M. Okamura, S. Chan, B. Hannaford, K. MacLean, and W. Provancher. Best Practices for Teaching Haptics, 2012. → pages 11

[57] S. A. Paneels, M. Anastassova, and L. Brunet. TactiPEd: Easy Prototyping of Tactile Patterns. *INTERACT '13*, 8118:228–245, 2013. doi:10.1007/978-3-642-40480-1. URL http://link.springer.com/10.1007/978-3-642-40480-1. → pages 12, 26

[58] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, 2nd edition, 1980. ISBN 0465046274. URL http://books.google.com/books?id=HhIEAgUfGHwC&pgis=1. → pages 97, 98

[59] M. Resnick, B. Myers, K. Nakakoji, B. Shneiderman, R. Pausch, T. Selker, and M. Eisenberg. Design principles for tools to support creative thinking. In *NSF Workshop Report on Creativity Support Tools.*, Washington, DC, 2008. URL http://medcontent.metapress.com/index/A65RM03P4874243N.pdf. → pages 13, 26

[60] P. Richer. A phenomenological analysis of the perception of geometric illusions. *Journal of Phenomenological Psychology*, 8(2):123–135, 1978. → pages 14

[61] J. Rovan and R. Hayward. Typology of Tactile Sounds and their Synthesis in Gesture-Driven Computer Music Performance. *Trends in Gestural Control of Music*, pages 297–320, 2000. → pages 17

[62] J. Ryu and S. Choi. posVibEditor: Graphical authoring tool of vibrotactile patterns. In *HAVE '08*, pages 120–125. IEEE, Oct. 2008. ISBN 978-1-4244-2668-3. doi:10.1109/HAVE.2008.4685310. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4685310. → pages 11, 26, 45, 46

[63] O. Schneider, S. Zhao, and A. Israr. FeelCraft: User-Crafted Tactile Content. In *Proceedings of AsiaHaptics 2014*, 2014. → pages 43, 44, 46

[64] O. S. Schneider and K. E. MacLean. Improvising Design with a Haptic Instrument. In *HAPTICS '14*, Houston, USA, 2014. → pages 15, 26, 45, 61

[65] O. S. Schneider and K. E. MacLean. Reflections on a WYFIWIF Design Tool. In *Workshop on Tactile User Experience Evaluation Methods at CHI 2014*, 2014. → pages 15

[66] D. A. Schön. *The Reflective Practitioner*. 1982. → pages 12, 13, 46

[67] H. Seifi, C. Anthonypillai, and K. E. MacLean. End-user customization of affective tactile messages: A qualitative examination of tool parameters. In *HAPTICS '14*, pages 251–256. IEEE, Feb. 2014. ISBN 978-1-4799-3131-6. doi:10.1109/HAPTICS.2014.6775463. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6775463. → pages 11, 43, 46, 61

[68] H. Seifi, K. Zhang, and K. MacLean. VibViz: Organizing, visualizing and navigating vibration libraries. In *World Haptics '15*, 2015. → pages 11, 44, 45, 46, 48, 56, 61, 63, 64, 65, 66, 67, 79

[69] J. Seo and S. Choi. Initial study for creating linearly moving vibrotactile sensation on mobile device. In *HAPTICS '10*, pages 67–70. IEEE, Mar. 2010. ISBN 978-1-4244-6821-8. doi:10.1109/HAPTIC.2010.5444677. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5444677. → pages

[70] B. Shneiderman. Creating creativity: user interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction*, 7(1): 114–138, Mar. 2000. ISSN 10730516. doi:10.1145/344949.345077. URL http://dl.acm.org/citation.cfm?id=344949.345077. → pages 12, 13

[71] C. Swindells, E. Maksakov, K. MacLean, and V. Chung. The Role of Prototyping Tools for Haptic Behavior Design. In *HAPTICS '06*, pages 161–168. IEEE, 2006. ISBN 1-4244-0226-3. doi:10.1109/HAPTIC.2006.1627084. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1627084. → pages 11, 26, 45, 46, 57

[72] C. Swindells, S. Pietarinen, and A. Viitanen. Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. In *HAPTICS '14*, pages 515–521. IEEE, Feb. 2014. ISBN 978-1-4799-3131-6. doi:10.1109/HAPTICS.2014.6775509. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6775509http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6775509. → pages 12, 26, 45, 46, 57

[73] D. Ternes and K. E. MacLean. Designing Large Sets of Haptic Icons with Rhythm. *Haptics: Perception, Devices and Scenarios*, 5024:199–208, 2008. URL http://www.springerlink.com/index/atl5686222k242m4.pdf. → pages 11, 45, 61, 63, 69

[74] A. Warr and E. O'Neill. Understanding design as a social creative process. In *Creativity & cognition - C&C '05*, pages 118–127. ACM Press, Apr. 2005. ISBN 1595930256. doi:10.1145/1056224.1056242. URL http://dl.acm.org/citation.cfm?id=1056224.1056242. → pages 12, 13, 43, 46, 51, 56

[75] H.-Y. Yao and V. Hayward. Design and analysis of a recoil-type vibrotactile transducer. *The Journal of the Acoustical Society of America*, 128(2): 619–627, Aug. 2010. ISSN 1520-8524. doi:10.1121/1.3458852. URL http://link.aip.org/link/?JASMAN/128/619/1. → pages 9, 17

[76] T. Yoo, Y. Yoo, and S. Choi. An Explorative Study on Crossmodal Congruence Between Visual and Tactile Icons Based on Emotional Responses. In *ICMI '14*, pages 96–103. ACM Press, Nov. 2014. ISBN 9781450328852. doi:10.1145/2663204.2663231. URL http://dl.acm.org/citation.cfm?id=2663204.2663231. → pages 9

[77] Y. Zheng and J. B. Morrell. Haptic actuator design parameters that influence affect and attention. In *2012 IEEE Haptics Symposium (HAPTICS)*, pages 463–470. IEEE, Mar. 2012. ISBN 978-1-4673-0809-0. doi:10.1109/HAPTIC.2012.6183832. URL http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6183832. → pages 11, 23