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)

July 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: r0.2

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 colleagues. 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 Chapters 1, 2, and 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, written with Dr. MacLean and PhD candidate Hasti Seifi.

In Chapter 3, Oliver contributed all work and ideas, with feedback and guidance from supervisor Dr. Karon MacLean. This work has been published as full conference paper Schneider and MacLean [91] and the associated demo **?**], and as workshop paper Schneider and MacLean [92]. The software has been released as an open-source project at https://github.com/ubcspin/mHIVE.

In Chapter 4, Oliver contributed most work and ideas, with initial interviews with designers and haptic experts conducted by Disney Research. This work was conducted while on internship at Disney Research Pittsburgh, with some supplementary work done at UBC, with feedback and guidance from internship supervisor Dr. Ali Israr and Dr. MacLean, and has been published as **?**] and the associated demo **?**].

In Chapter 5, Oliver contributed all work and ideas, with feedback and guidance from supervisor Dr. MacLean. This has been published as **?**] with the associated demo]; the software Macaron has been released as an open-source project at https://github.com/ubcspin/Macaron and is available online at http://hapticdesign.github.io/macaron. As of this writing, subsequent development of the core Macaron tool and MacaronMix includes work by Matthew Chun, Benson Li, Ben Clark, and Paul Bucci.

In Chapter 6, Oliver was part of a collaborative team together with PhD candidate Hasti Seifi, undergraduate summer student Matthew Chun, and master's student Salma Kashani, all supervised by Dr. 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 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 Schneider et al. [90].

In Chapter 8, Oliver contributed interview transcription, qualitative analysis, writing, organization and analysis of the HaXD'15 workshop (http://oliverschneider.ca/HaXD/) with guidance from Dr. MacLean. UBC alumni Dr. Colin Swindells conducted interviews and provided interview notes and initial analysis ideas, with supervision (?) by Dr. MacLean and Dr. Kellogg Booth. Drs. MacLean, Swindells, and Booth all provided feedback during writing. As of this writing, this paper has been submitted to International Journal of Human-Computer Studies, Special Issue on Multisensory Human-Computer Interaction 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, yet 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. 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, both tactile (skin-based) and proprioceptive (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 (Chapters 3-5).** 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 (Chapters 6 and 7).** 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 (Chapter 8).** 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.

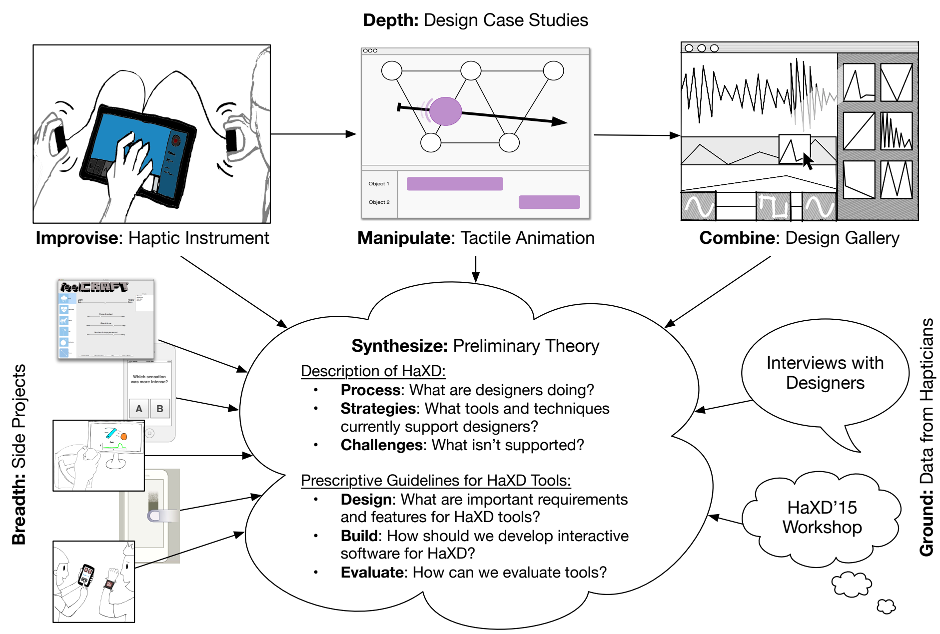


Figure 1.1: Approach overview. Three case studies investigate VT tools in-depth; findings are synthesized with side projects and grounded data into a preliminary theory.

1.2.1 Depth: Vibrotactile Design Tool Case Studies

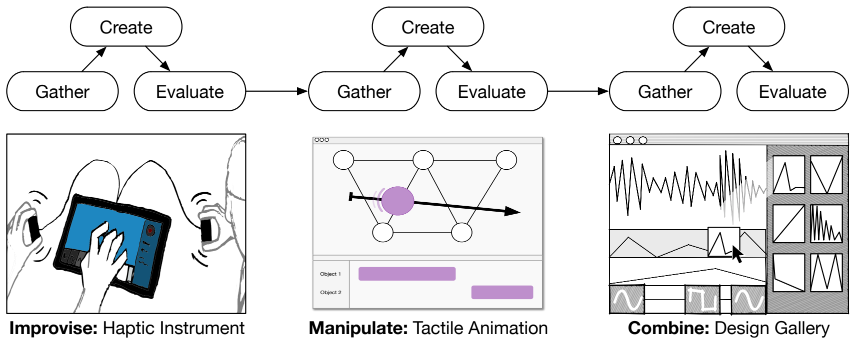
Each case study investigates a different set of design concepts with varying user populations, VT device, and design challenges (Figure 1.2), but restricts scope to

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.

Initial Exploration: The Haptic Instrument (Chapter 3) 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. We also found informal, collocated collaboration useful, but leave future examination of collaboration support to side projects (described next in Section 1.2.2).

Direct Manipulation Pipeline: Tactile Animation (Chapter 4) In Study 2, Tactile Animation, we developed a single abstracted animation object directly manipulated in both space and time. In this study, we focused on building a usable tool to support exploration and refinement, and investigate a generalized rendering pipeline in detail to understand how to build haptic design tools. Animators found our tactile animation tool, Mango, easy-to-use, and confirmed our findings about the value of real-time exploration. We also found that "soft features", like copy/paste and undo/redo, were extremely important.

Example Use and Analytics: Macaron (Chapter 5) 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, we 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. Macaron was implemented using the understanding we gained from Study 2, giving us more opportunity to focus on capturing and studying the design process, especially using interaction logs to investigate example use. 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.



	Improvise	Manipulate	Combine
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.

1.2.2 Breadth: Focused Haptic Design Projects

Each case study provides concrete knowledge for building a VT authoring tool, and some insight into the VT design process. However, VT technology can be used in many different scenarios, and there are many devices and experiences that involve other haptic modalities. To broaden our understanding of haptic design, we undertook several more focused haptic design projects to look at different activities, application areas, and haptic modalities.

In **Chapter 6**, we investigate in-depth a technique for large-scale feedback:

Feedback at Scale: HapTurk is a technique for collecting large-scale feedback on VT icons. Haptic devices cannot be sent to hundreds or thousands of people for feedback, but collecting in-lab feedback can be expensive, and informal feedback from colleagues is limited in scope. We investigate whether visual or low-fidelity *proxies* can stand in for high-fidelity VT effects.

In **Chapter 7**, we describe several smaller projects that gave opportunities for practicing haptic design and exploring other types of haptic feedback:

FeelCraft (Section 7.1) is a plug-in architecture that augments media with customizable spatial VT effects. We use FeelCraft to explore existing infrastructure for haptic media, and to design VT effects for a popular video game, MineCraft.

Feel Messenger (Section 7.2) is a chat program augmented with expressive, customizable VT effects using commodity hardware and APIs.

RoughSketch (Section 7.3) is a painting application for the TPad Phone, a variable-friction mobile device, for the World Haptics 2015 Student Innovation Challenge. 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.

HandsOn (Section 7.4) is a conceptual model for creative education software using low-cost, DIY haptic hardware, giving us an understanding of how to work with 1-degree of freedom force feedback and an educational context.

CuddleBit (Section 7.5) is a project inspired by the Haptic Creature [] and CuddleBot project []. We use small, breathing robots to explore the display of emotion, and extend our findings from VT design tools into new tools for this modality: Voodle and MacaronBit.

1.2.3 Ground: Data from Haptic Experience Designers

⌈**OS** *TODO: Update this once I've gone over Ch8 again*⌋ 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:

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?

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

⌈**OS** *I've already outlined what I'm talking about. What else can I put here to set up the dissertation?*⌋ ⌈**OS** *TODO: unite this text with some of the previous descriptions, somehow.*⌋ 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, we provide the relevant background for this dissertation. We begin with an overview of haptic technology and perception. Next, we discuss the application space for haptics and why haptic experiences are increasingly important to design. We then discuss the previous work in HaXD and related support tools, identifying why this is an area for improved understanding. After, we discuss non-haptic creativity support tools and design theory which provided inspiration and guiding principles. Finally, we present the qualitative and quantitative methodologies used in this dissertation. Throughout the chapter, we contextualize this work and HaXD in both the haptics and HCI communities.

2.1 An Overview of Haptic Technology and Perception

The term "haptic" was coined by German researcher Max Dessoir to refer to the study of touch, in a similar to "optic" for sight and "acoustic" for sound [**?**]. Today, it refers to both the study of the psychology and perception of the senses of touch, and frequently to technology that employs touch as a method of feedback. Haptic technology is typically separated into two classes based on the main sense modality: *tactile* sensations, and *proprioception*, or the sense of body location and force. These two types of feedback are useful for different purposes, e.g., people use their fingerpad's tactile senses to derive texture, while but kinaesthetic feedback can help to infer weight [56]. However, they are often combined for more convincing results [81]. An excellent overview of the haptic senses is available by Lederman and Klatzky [57], and a practical introduction to the technology is available by Hayward and Maclean [39]. Here, we focus our coverage on the sensations directly studied by this dissertation, while also portraying the diversity of haptic experiences and technology.

2.1.1 Tactile Perception and Technology

⌈**OS** *Goal: Explain the main mechanisms we use in the dissertation's studies, persuade the reader that I know enough about tactile perception, and show the variety of tactile displays and experiences.*⌋ Tactile sensations rely on multiple sensory organs in the skin, each of which detect different properties, e.g., Merkel disks detect pressure or fine details, Meissner corpuscles detect fast, light sensations (flutter), Ruffini endings detect stretch, and Pacinian corpuscles detect vibration [**?**]. Of these, the Pacinian corpuscle is most widely targeted by technology through vibrotactile (VT) sensations, where vibrations stimulate the skin. VT sensations are accessible, well-studied, and increasingly widespread, and can be passively felt, easing implementation. Our in-depth design tool studies thus focus on VT experiences.

VT actuators can take may forms. Eccentric mass motors (sometimes "rumble motors" or, when small, "pager motors") are found in many mobile devices and game controllers, and are affordable but inexpressive. More expressive are tactors, or voice coils, implemented in a variety of ways and offering two degrees of freedom: frequency and amplitude. Piezo actuation is a very responsive technique that is typically more expensive. While tactors directly stimulate the skin, linear resonant actuators (LRAs) shake a mass back and forth to vibrate a handset in an expressive way; a common research example is the Haptuator [110]. Instead of directly stimulating the skin, this actuator typically shakes another device held by the user, such as a mobile device [111] or pen [20]. As of writing (2016), this type of actuator is increasingly deployed in consumer products (e.g., Apple's Taptic Engine).

Actuators like VT devices can be used alone or put together spatial multiactuator displays like seats [45**?**], belts [84**?**], wristbands [84**? ?**], vests [51**?**], and gloves [54**?**]. These can be arrange into grids, either dense tactile pixels ("taxels") [54] or sparse arrays [45**?**], to provide explicit 2D output on a plane. Multiactuator arrays increasingly used in conjunction with tactile illusions.

As with visual displays, like the illusion of motion with high framerates, illusions play a valuable role in haptic sensations [38]. Some effects are influenced by other senses. In the classic size-weight illusion [**?**], when two weights have the same mass but different sizes, the smaller is perceived to be heavier, whether size is seen or felt [38]. A striking, recent example is the use of visual dominance to use a single physical block to provide haptic feedback for multiple virtual blocks by distorting the visual position of the user's arm [**?**]. We employ similar techniques in our FeelCraft and Feel Messenger projects, using visual feedback to prime users to haptic sensations (Chapter 7).

Other illusions are purely tactile and useful for grid displays. Phantom tactile sensations [1], create illusory vibrations in between two or more VT actuators, opening up the space in-between actuators for display. Continuous motion can be simulated, e.g., Seo and Choi [96] created a perceived motion flow between two VT actuators mounted on the ends of a handheld device by controlling their intensity. Similarly, Lee et al. [63] created across-the-body and out-of-the-body illusions on a mobile device using up to four linear resonant actuators; **?**] used interpolation on a VT wristband for new interaction techniques. The Tactile Brush algorithm [44] combined phantom tactile sensations and apparent tactile motion to render high-resolution and moving haptic patterns on the back using a coarse grid of VT actuators. Other spatio-temporal VT illusions such as the "cutaneous rabbit" [101], where carefully timed discrete tactile stimuli create perceived motion, and Tau and Kappa effects [37, 38], where perceived distance between stimuli depending on their timing, can also be used with VT arrays. Similar illusions are possible using other tactile modalities, including temperature displays [**?**] and electrocutaneous stimulation [**?**]. We extend phantom sensations to 2D interpolation (e.g., between 3 actuators) to enable Tactile Animation (Chapter 4).

Another emerging tactile feedback mechanism is programmable friction. Surface friction, for example on a mobile touch screen, can be manipulated by both mechanical motion or electrical adhesion. The TPad [109] vibrates a plate at ultrasonic frequencies to create a cushion of air between the surface and the user's finger. This effect is programmable, and can be used to with a number of interactive scenarios [64]. Other techniques like electrovibration, deployed in TeslaTouch [3], and electrostatic forces [73] can create a similar effect. Strong electroadhesion [**?**] has the potential to create even larger shear forces, but comes with a high power cost. In RoughSketch (Chapter 7), we design for a mobile version of the TPad deployed on Android devices, the TPad phone (www.thetpadphone.com).

There are many other types of tactile stimulation used in haptic experiences. 2-dimensional pin-based grids like Optacon [**?**] and HyperBraille (www.hyperbraille.de) can display Braille and 2D images to the blind and visually impaired, and operate as a generic computer display [**?**]. Similar multi-point displays have been deployed on mobile devices. Edge Haptics uses dozens of linearly-actuated pins on the edge of a mobile device for tactile stimulation, similar to a a-dimensional braille pin display [**?**], while laterally moving pins can use skin-stretch as a display mechanism [66]. Electrocutaneous stimulation, by activating electrodes directly on the skin, has been effectively deployed for spatial tongue displays [**?**]. Temperature displays exploit warm and cold receptors in the skin for display, using Peltier junctions [**?**].

Proprioception and Force Feedback – ⌈**OS** *Goal: Explain the main mechanisms we use in the dissertation's studies, persuade the reader that I know enough about proprioception, and show the variety of force-feedback displays. This sec-*

tion will be smaller than the tactile section, and should make the distinction of verisimilitude and veridicality a bit clearer, because with FF we often have a virtual environment and are more concerned with passivity, stiffness, etc.⌋

Proprioception, the sense of force and position, is synthesized from multiple sensors as well: the muscle spindle (embedded in muscles), golgi-tendon organ (GTO) in tendons, and tactile and visual cues [52]. We distinguish proprioception from the related term kinaesthetic by being the general, synthesized sense, where kinaesthetic sensation is strictly the sense of motion ⌈**OS** *todo: make sure this is correct*⌋. Force displays are common in precise, specialized applications like robot-assisted surgery [82] or realistic sensorimotor training environments [104].

Force-feedback devices might have degrees of freedom of feedback (DoF), the number of forces or torques they can display. These devices render a *virtual environment*, with simulated forces depending on the input from the user. Common consumer-facing 3-DoF devices include the Geomagic Touch (previously the Sensable PHANTOM) and Falcon devices, offering force in three directions. 2-DoF designs like the pantograph [**? ?**] can provide displays on screens, walls, and tables. These displays have previously input on realistic simulation and rendering: e.g., making free space feel free, providing stiff virtual objects and walls, and avoiding saturation [72]. Open-hardware, self-assembled versions of these devices, such as WoodenHaptics [31] for 3-DoF devices and Haplet [32] for 2-DoF displays, have the potential to make haptics more accessible. Much previous work has been done on handling technical concerns, e.g., displaying complex polygonal objects with a "God object" [**?**], coordinating remotely situated devices or shared environments [**?**], and improving collision realism with transient forces [55]. More complex environments are primarily programmed in using APIs like CHAI3D, OpenHaptics, or Unity.

Another approach is to use simple force feedback, especially for haptics education [50]. 1-DoF devices include linear actuators pushing on the user and haptic knobs, e.g., the UBC Twiddler [28, 70, 97], and paddles, e.g., the HapKit [**?**]. The UBC SPIN lab has also adopted 1-DoF force feedback in its affective robot, the Haptic Creature [**? ?**], the CuddleBot, and CuddleBits [**?**]. We explore force-feedback design with the HapKit and CuddleBits in Chapter 7.

2.2 The Value of Haptic Experiences

2.2.1 Education

Haptic technology has the potential to improve educational resources, especially to those lacking resources. Montessori methods have long espoused the value of physical learning aids, especially using physical *manipulatives* [**?**]. There is evidence to support these techniques: in a meta-analysis of 55 studies, **?**] found that physical manipulations improve several learning outcomes, with influence by other instructional variables. Studies of gestures have also found value in students "being the graph" by physically acting out mathematical shapes, grounding abstract knowledge in embodied experience [**?**]. These techniques have roots in constructivist learning, where learners use existing understandings as a *transitionary object* to understand new concepts [85].

Haptic technology is well-positioned to support embodied learning, and there is early evidence for its efficacy. Haptic feedback has been shown to improve temporal aspects when training motor skills [**?**]. In a study for molecular chemistry education, **?**] found students had higher test scores when they interacted with their haptic learning interface; students reported engagement. In Chapter 7 we describe results from an early learning interface for low-cost haptic displays [**?**], showing that haptic technology can improve engagement and make lasting impressions.

Active learning and creativity in education??

2.2.2 Immersive Media

A popular application for haptic experiences is augmented, immersive media experiences. Actuated tactile feedback has been used as early as 1959 in the movie *The Tingler* [43]. 4D theatres and theme park rides use bursts of air or water sprays to engage the audience. Companies like D-Box (www.d-box.com) augment films with haptic tracks that both low-frequency movements and high-frequency vibrations, and can be found in theatres across the world. Buttkicker (www.thebuttkicker.com) also augments 4D theatres, and provides products for home theatre setups.

Haptic experiences are also increasingly of interest in virtual reality (VR) environments. Skin stretch techniques, explored in [**?**] and now commercialized by Tactilcal Haptics (tacticalhaptics.com), augments virtual-reality setups by simulating forces and torques using handheld controllers, lending stronger immersion for virtual environments and VR games. Haptic Turk [**?**] and TurkDeck [**?**] are innovative explorations of high-fidelity haptic experiences in virtual environments using people as actuators. Impacto uses electrical muscle stimulation and a solenoid actuator to create wearable haptic feedback with both kinaesthetic and tactile feedback [65]. haptic retargetting [**?**]

Previous work has also attempted to add greater immersion to broadcast media by including haptic sensations. Modhrain and Oakley [74] present an early vision of Touch TV, using active touch with two-DOF actuators embedded in remote controllers. Gunther et al. [35] studied a full-body vibrotactile suit to create music-like "cutaneous grooves", helping to identify the artistic space of VT sensations, including concerts with tactile compositions. More recently, the proliferation of online streaming video has developed opportunities to add haptic sensations using novel data structures. Tactile Movies [54] looks into augmenting movies with spatial VT feedback, including an authoring interface. The Haptic Application Meta Language (HAML)[**?**] is an XML-based data format for adding haptics to MPEG-7 video, eventually augmented with the HAML Authoring Tool (HAMLAT) [**?**]. **?**] eventually adapted an XML approach to YouTube, and Gao et al. [33] developed related online MPEG-V haptic editing.

Augmented media experiences, or haptic-audiovisual (HAV) content [22], have used different methods of input. One approach is to use camera motion sourced from accelerometers [**?**] to actuate audience members' hands and head in a HapSeat [21**?**]. Later editable with H-Studio [**?**], this work has proposed the concept of Haptic Cinemotography [23], including basic principles of composition when combined with video [**?**]. Other approaches include automatic conversion of audio content Several studies have looked into automatic conversion from audio streams [13, 61**?**] or video streams [53] to VT or force-feedback output.

⌈**OS** *todo: Haptically Annotated Movies: Reaching Out and Touching the Silver Screen, Gaw 2006*⌋

2.2.3 Affective Computing

Haptic therapy -¿ creature, etc. [**? ? ? ?**]

Communicating emotions Social? Chang 2002 ComTouch Brave 1997 inTouch Haans2006 Mediated Social Touch Hook 2008

Ciollaboration

2.2.4 Accessiblity

Sensory substitution, first pioneered by Bach-y-Rita [**?**], is an impressive technique often using haptic senses to augment or replace other sense. A wide variety of devices have been developed and studied for the visually impaired; see [**?**] for a recent survey and discussion of user preferences.

Guidance [**?**] back display for guidance, Linderman2005 [**?**] [**?**] [**?**]

mobile? [**?**], turn taking [12]

2.2.5 Medical

Dental and/or physical therapy?

2.3 Previous Efforts for Haptic Experience Design

⌈**OS** *Should we do a separate section for Creativity and Design itself? Or is that a subsection?*⌋

2.3.1 Editors and design tools

As long as designers have considered haptic effects for entertainment media, they have needed compositional tools [35].

Custom editors (such as D-Box Motion Code Editor) and software plugins are provided to media designers that overlaid the visual and audio content with haptics, and allow designers to generate, tune and save frame-by-frame haptic content in an allocated track for it to play simultaneously with the media content.

By tuning parameters of these effects, users could personalize haptic content, embed it in games and share effects with other users. Similar devices and authoring schemes are also developed for online social interactions using custom multi-actuator haptic devices [54, 84, 103].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 [28], Haptic Icon Prototyper [99], and posVibEditor [89] use graphical mathematical representations to edit either waveforms or profiles of dynamic parameters (torque, frequency) over time. The Vibrotactile Score [62] was shown to be generally preferable to programming in C and XML, but required familiarity with musical notation [60]. The Demonstration-Based Editor [41] 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 [100].

The control of multi-actuator outputs has been explored by TactiPEd [84], Cuartielles' proposed editor [19], and the tactile movie editor [54]; 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 **??**.

2.3.2 Platforms?

There are 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 [20]. The HapticTouch Toolkit [58] and Feel Effect library [46] 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 [34]. More recently, Wooden Haptics gives open-source access to fast laser cutting techniques for force feedback development [31], and faBrickation streamlines prototyping for 3D printing [78]. 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.

2.3.3 Language of touch ⌈OS *and schema?*⌋

Some higher-level perspectives offer outcome targets or design attitudes to guide haptic practitioners. "DIY Haptics" categorize feedback styles and design principles [39, 68]. "Ambience" is proposed as one target for a haptic experience [69]. Haptic illusions can serve as concise ways to explore the sense of touch, explain concepts to novices and inspire interfaces [37]. "Simple Haptics", epitomized by *haptic sketching*, emphasizes rapid, hands-on exploration of a creative space [75, 76]. Haptic Cinematography [23] uses a film-making lens, discussing physical effects using cinematographic concepts. The notion of distributed cognition [42] 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 [50, 83].

Haptics has often made use of metaphors from other fields. Haptic icons [67], tactons [5], and haptic phonemes [29] are small, compositional, iconic representations of haptic ideas. Touch TV [74], tactile movies [54], haptic broadcasting [11], and Feel Effects [46] 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 [60, 62]. Other musical metaphors include the use of rhythm, often represented by musical notes and rests [7, 9, 12, 102]. Earcons and tactons are represented with musical notes [5, 6], complete with tactile analogues of crescendos and sforzandos [8]. The concept of a VT concert found relevant tactile analogues to musical pitch, rhythm, and timbre for artistic purposes [35]. Correspondingly, tactile dimensions have been also been used to describe musical ideas [25].

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 [28, 80]. Language is a promising way of capturing user experience [79], and can reveal useful parameters, e.g., how pressure influences affect [112]. Tools for customization by end-users, rather than expert designers, are another way to understand perceptual dimensions [94, 95]. However, this work is far from complete; touch is difficult to describe, and some even question the existence of a tactile language [48].

2.4 Non-Haptic Design and Creativity Tools

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

2.4.1 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 [106]. Also known as the "problem setting" [93], "analysis of problem" [106], or "collect" [98] 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 [93]. 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 [10, 40], which we explore in **??**.

2.4.2 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 [42]. 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 [10]. 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 [36, 86], while in groups, sharing multiple designs improves variety and quality of designs [24].

Sketching supports ideation, evaluation, and multiple ideas, allowing the designer to explicitly make moves in a game-like conversation with the problem [93]. It is so important that some researchers declare it to be the fundamental language of design, like mathematics is the language for scientific thinking [16]. 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" [86].

2.4.3 Collaboration

Design is a collaborative process with the potential for generating more varied ideas [106], and is important for creativity support tools [86, 98]. Although sometimes group dynamics influence the design process negatively, proper group management and sharing of multiple ideas results in more creativity and better designs [40]. Shneiderman in particular has championed collaboration in design [98], 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) [27]. 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 **??**).

2.5 Methodology

Studying design process, creativity support tools, and haptic sensations is challenging and requires a robust methodology. Our research questions seek to describe the process, strategies, and experience of HaXD, and to inform the design of supporting tools. Our approach is to use mixed methods, as appropriate for our research questions. We begin by using qualitative techniques to gather rich, generative data from design tools and design processes, and to inform iteration. Through our work, we increasingly complement this data with quantitative methods, moving towards large-scale data collection for our generated theories with deployed tools.

2.5.1 Inspiration and Perspectives

extended mind and embodiment Distributed cognition, embodiment, extended mind... ??

2.5.2 Qualitative methodologies

In this dissertation, we draw from the philosophical and methodological traditions of phenomenology, and the methodology of grounded theory.

Phenomenology is both a philosophical tradition and a social science methodology based upon that tradition that involves the study of subjective experience. We use Moustakas [77] as our primary guide through both, as it focuses on practical methodological concerns but provides a strong philosophical background; Creswell [15] provided an overview of various methodologies and resources for phenomenology. Critical components include *horizonalization*, or preparing oneself to consider all of the participant's statements equally and with fresh eyes; distinguishing and synthesizing *textural* descriptions, e.g., the participant's verbatim explanation of the experience, with *structural* description, or the analytical interpretation through psychology (or HCI) theories; and the documentation of the researcher's own experience with the phenomenon of study. Phenomology as a methodology has been used in psychology to investigate topics ranging from visual illusions to tactile experience [15, 79, 87].

Methodologies, like phenomenology, often include a set of methods combined with their philosophical and epistemological underpinnings. In this work, we specifically use the Stevick-Colaizzi-Keen methods as described by Moustakas [77]. 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 through affinity diagrams, writing and re-writing of thematic descriptions, and reflection guided by phenomenological philosophy. We use this technique exactly in our first exploratory study (Chapter 3), and later combine it with Grounded Theory methods.

Grounded Theory is another well-known methodology first described by **?**]. We adopt the more flexible methodology described by Corbin and Strauss [14], as it allowed us to integrate with our phenomenological methods. We principally adapt the methods used in Grounded Theory, specifically, memoing (writing about each focused quotation or MU), constant comparison (comparing each new memo and codes to previous ones), and open and axial coding (creating codes, or concepts, linking them together, then categorizing statements or observations based on codes). This technique especially facillitated video analysis in Chapter 4 and Chapter 5, and allowed for quantitative count-based data and simple statistics to complement our interview-based findings.

We also note that researchers who use methods from the umbrella term "qualitative research" often blend techniques from many methodologies. Phenomenology and grounded theory are two such methodologies, others include hermeneutics, ethnography [77], ethnomethodology, and thematic analysis [**?**]. For example, ethnography introduces the concept of "thick description" [], where the research tries to use detailed, evocative language to convey a rich sense of being in the observed environment. This technique is used more generally in observational notes and when writing up qualitative results to provide the rich data studied by qualitative methods.

2.5.3 Towards quantitative methods at scale

While some design scholars adopt qualitative techniques [17, 18, 93], others have developed quantitative techniques. When studying graphic design for ads, Dow et al. [24] used ratings by experts as well as click-through rates and other online analytics for actual deployed ads from their study. **?**] used MTurk to generate sketches of aliens in several conditions (of exposure to examples), then deployed another MTurk task to label each drawing with features like antennae or feet. Lee et al. [59] had end-users rate graphic designs in both an in-lab study and over Mechanical Turk, and recorded time participant designers spent on each component. These approaches allow for hypothesis testing for specific research questions, but require infrastructure unavailable to haptics, notably, crowd deployment, rating scales, and mature input tools. In our early work, we found qualitative feedback to be sufficient while developing our understanding of how to build design tools. Later, we begin to approach this infrastructure, using an online editor with analytic logs in Macaron (Chapter 5), and examining the potential for crowdsourced feedback with HapTurk (Chapter 6). While a valuable goal, large-scale quantitative feedback on HaXD remains outside the scope of this dissertation.

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 [91] and at a CHI 2014 workshop [92].

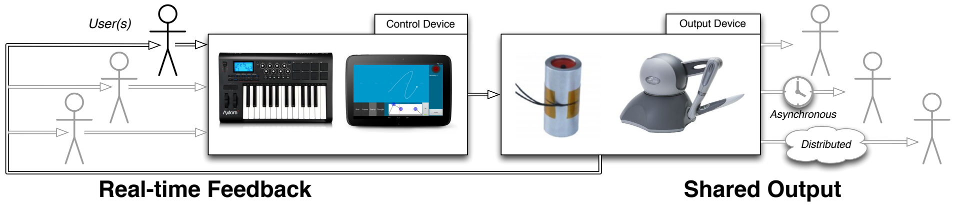


Figure 3.2: The haptic instrument concept. One or more people control the instrument, and receive real-time feedback from the device. Any number of audience members can feel the output in real time as well. Control methods can vary, from traditional musical control devices (such as the M-Audio Axiom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

playback. Using a music composition metaphor (as in [62]), 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 [110] 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 [5, 8, 9, 35, 88]. mHIVE is implemented in Java using the Android SDK [2], and the FMOD sound synthesis library [30] to produce sounds, sent to two or more Haptuators through an audio jack. We deployed mHIVE on an Android Nexus 10

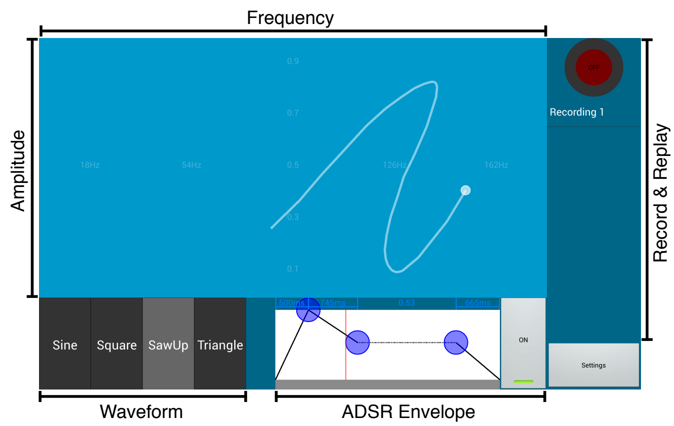


Figure 3.3: mHIVE interface. Primary interaction is through the amplitude-frequency view, where visual feedback is provided through a circle (current finger position) and a trail (interaction history).

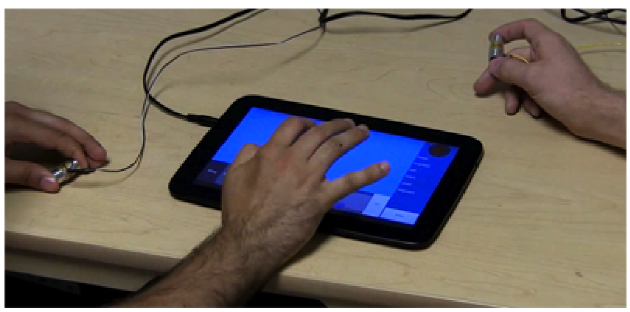


Figure 3.4: Study setup. Both the participant (left) and the interviewer (right) feel the same sensation as the participant controls mHIVE.

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 [77]. 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 [26], 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 [14]) 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

[15], 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 [4] 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 [18]. 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 [76]. 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 [60, 62] or the hapticon editor [28]. 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 [48]. 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 [112]. The heavy use of onomatopoeias is reminiscent of Watanabe *et al.*'s work with static materials [107]. 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 [79]), 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

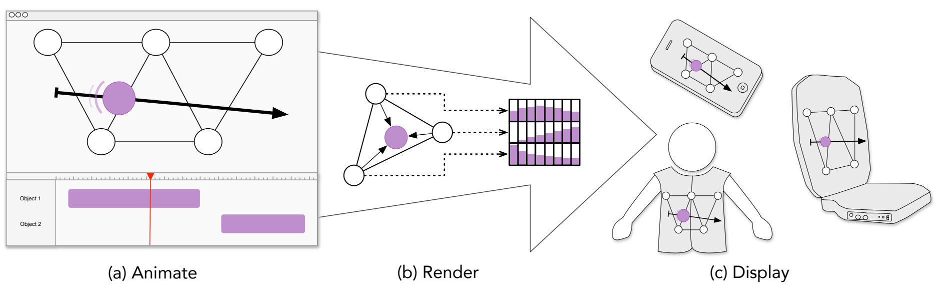
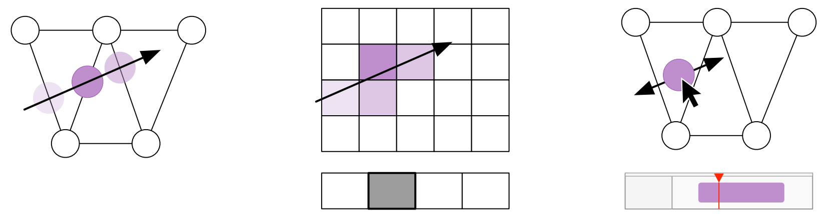


Figure 4.1: Concept sketch for tactile animation. An artist draws an animated sequence in the user interface and the user experiences phantom 2D sensations in-between discrete actuator grids.

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.

Tactile Animation let us ⌈**OS** *TODO: prosify this additional framing, outline*

[1]Published and demoed at UIST 2015 [**?**].



(a) Tactile Brush [44]: precomputed paths

(b) Tactile Video [54]: frames of tactile pixels

(c) Tactile Animation: direct manipulation

Figure 4.2: Comparison between related systems.

chapter.] - develop a full pipeline editing tool - ground our designs in industry concerns - evaluate with non-haptic media designers to investigate transfer effects - expand our design palette to spatial vibrotactile effects, unexplored directly in the haptic instrument case study - in particular, we expand to a directly-manipulated metaphor - look at device generality in a different way than the haptic instrument. In the haptic instrument, one can build a haptic instrument for new output devices by connecting nput parameters to different output devices. Here we look at a class of devices - 2D vibrotactile grids, but possibly other 2D grids. - this direct manipulation metaphor is distinguished from other spatial control by Figure 4.2.

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

LR	Description
LR1	**Real-Time Playback** [76, 91] 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** [49, 86, 91] 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** [28, 40, 84, 99, 100] 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** [54, 60, 61, 84] 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** [28, 84, 89, 99, 100] 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** [19, 54, 84] 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** [54, 76, 100] Haptic perception depends greatly on additional senses [37]. By providing audio and visual feedback, these effects can be mitigated and the designer can experience haptic sensations in context.
LR8	**User Feedback** [91, 100] 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.

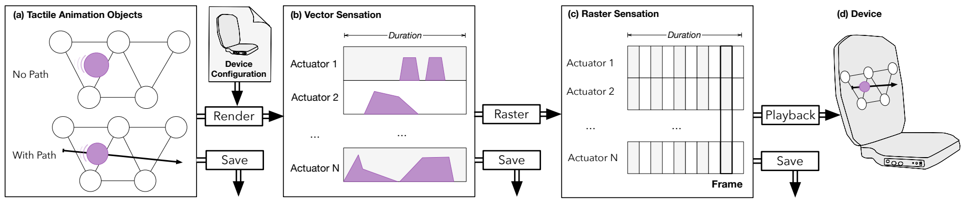


Figure 4.3: Tactile animation rendering pipeline. Users can: (a) create tactile animation objects; (b) render objects to actuator parameter profiles (such as amplitude) with our rendering algorithm; (c) rasterize vector sensations into frames; (d) play the sensation on the device.

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 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

⌈**OS** *begin needs-summary*⌋ 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, 44]. 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 [63**?**]; 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.

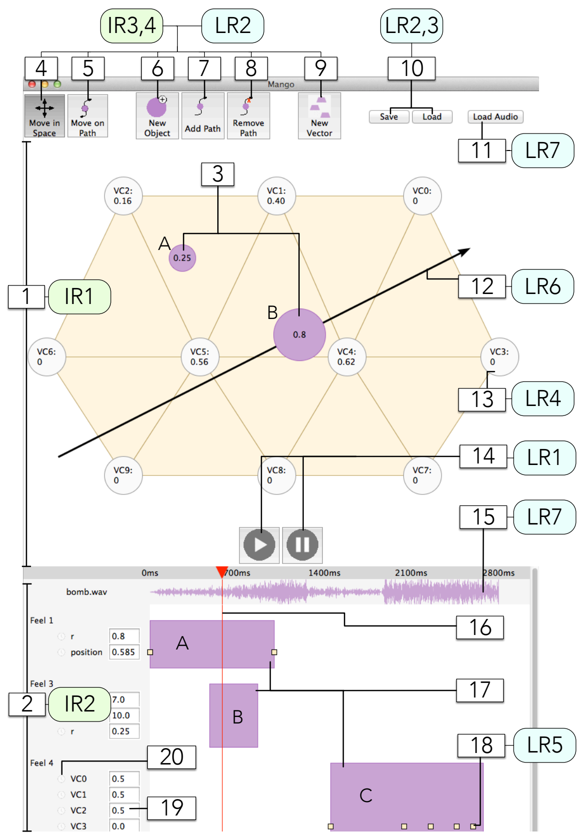


Figure 4.4: Mango graphical user interface. Key components are labeled and linked to corresponding design requirements.

Vector formats are similar to those in previous work (e.g., [28]). Instead of objected-based definitions, as in tactile animation objects, parameters are defined 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 [44].

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. ⌈**OS** *end needs-summary*⌋

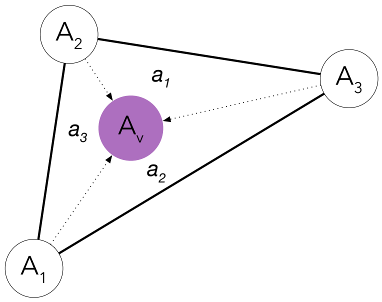
4.2 Rendering Algorithm

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., [44]). 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



(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 to determine physical actuator output (A_{1-3}) from virtual actuator intensity (A_v) and barycentric coordinates (a_{1-3}).

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

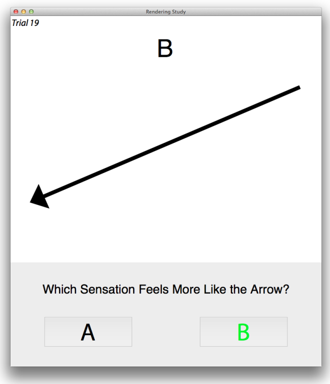
In the linear interpolation model, barycentric coordinates are linearly related to actuation amplitude. In the log model, these coordinates are scaled logarithmically, as perceived intensity is related to physical vibration amplitude [105]. In the power model, coordinates are coupled to the power (square of the amplitude) of vibrating stimulations [105]. 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 [44] 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.,



(a) Rendering study interface

(b) Output device with highlighted actuators

Figure 4.6: Rendering study setup and user interface.

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 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 $\sim$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

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.

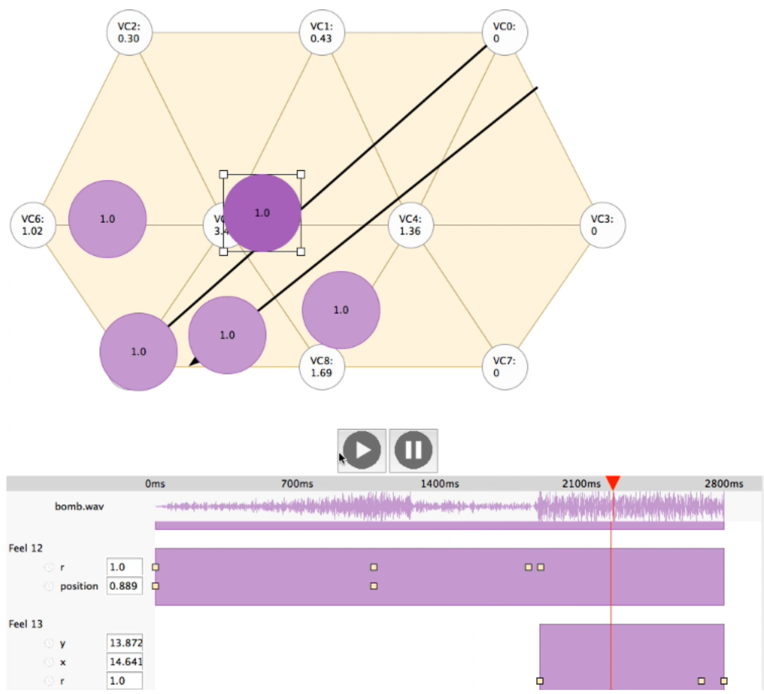


Figure 4.7: Example of P2’s animation for matching a sound. See the accompanying video for all participant animations.

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 [14] informed by phenomenological protocols [77]. Analysis resulted in four themes.

Theme 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).

Theme 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).

Theme 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).

Theme 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

⌈**OS** *TODO: restructure discussion when other chapters are more populated, to connect it better.*⌋ 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.

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.

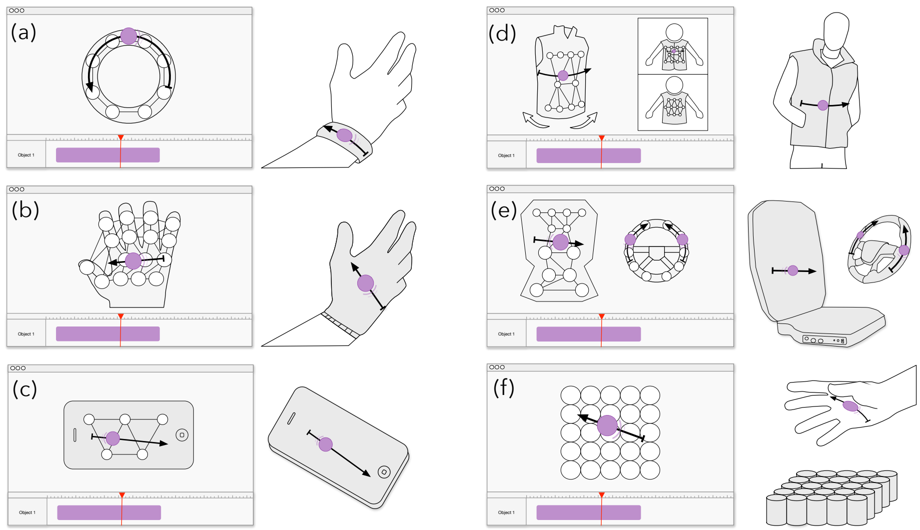


Figure 4.8: Tactile animation could define motion with (a) 1D actuator arrays, (b) dense and sparse VT grids, (c) handhelds, (d) 3D surfaces, (e) multi-device contexts, and (f) non-VT devices like mid-air ultrasound.

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 4.8d): 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 4.8e): 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 4.8f): 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 [108]; this sensation could be manipulated through a tool like Mango. Similarly, passive force-feedback sensations (e.g., Hapseat [21]) 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 [53], 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 4.8e).

4.4.5 Demo

⌈**OS** *Add image, description, takeaway from demo*⌋

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

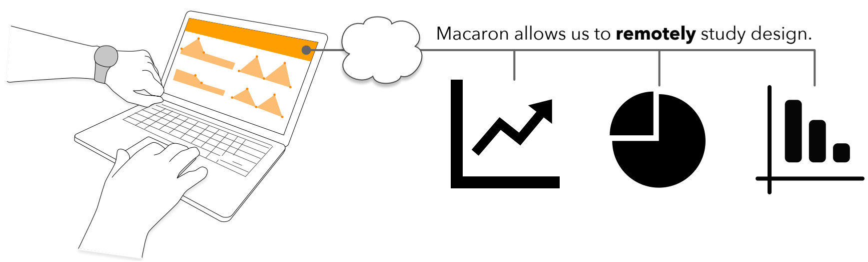


Figure 5.1: Concept sketch for a Macaron, an online, open-source VT editor features incorporable examples and remote analytics.

In both the Haptic Instrument and Tactile Animation case studies, participants drew from their experience or external examples and requested features for repetition. Because we explored haptic design tool implementation and rendering pipeline in-depth in Chapter 4, with Project Macaron[1] we instead move the focus to capturing more of the design process, with example-use as the target phenomenon. Creativity often sparks when an inventor, examining existing ideas, sees a way to combine them with a novel twist [106]. An environment rich with *examples* is fuel for this fire. In industrial and graphic design [10, 40] their use improves process and final results [24, 59]. Design galleries are used in graphics and web design to facilitate the use of examples [59, 71]. 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.

[1]Published and demoed at Haptics Symposium [**?**].

Problem preparation – also known as the "problem setting" [93] or "analysis of problem" [106] step of design – involves immersion in the challenge and drawing inspiration from previous work. Both may come from the designer's experience, *repertoire* [93] 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 [10, 40]. Industrial designers collect objects and materials; web designers bookmark sites [40]. In graphics and web design, *design galleries* organize examples to be immediately at hand [59, 71]. Example-based tools often use sophisticated techniques to mix and match styles and content [**?**]: this requires immediate access to the examples' underlying structure.

Several effect libraries are available to designers of vibrotactile (VT) sensations, e.g., for accessible wayfinding [**?**] or media experiences [20, 46, 90**?**]. But despite the need for effect customizability [94], VT library elements are generally opaque in construction and immutable. Recent advances include limited parameter adjustability [46, 90] and faceted library search and browsing [95]. 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* [59, 71] (Figure 5.2) for VT sensations, then asked users to compare versions (Figure 5.3) that vary in example accessibility via *visibility* and *incorporability*, as they create VT effects for animations (Figure 5.4).

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.

5.1 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 5.2,5.3). This enables easy extensions to other devices, discussed in Chapter 7.

Tracks are the accepted language of temporal media editors (video, audio, and past haptic efforts [28, 89, 99]). We provide tracks for perceptually important "textural" parameters (amplitude and frequency); the user accesses periodic and

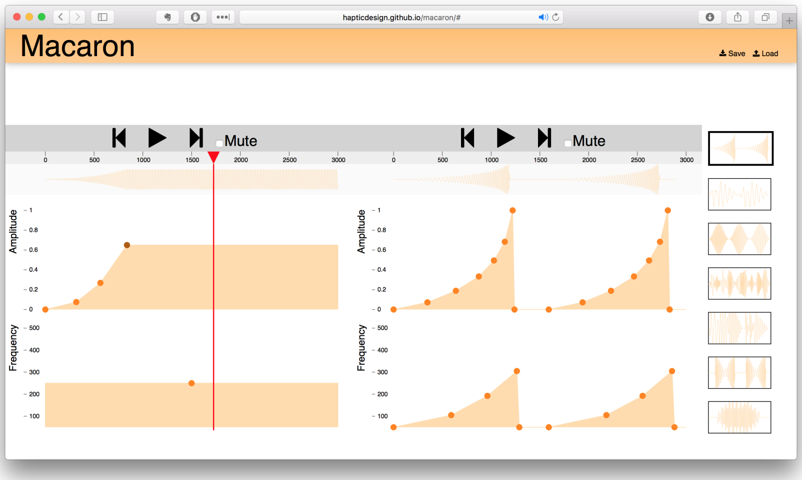


Figure 5.2: Macaron interface, "`hi`" version featuring both composability (copy and paste), and visibility of underlying parameters. The user edits her sensation on the left, while examples are selected and shown on the right. Macaron is publicly available at `hapticdesign.github.io/macaron`.

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 [59, 71**?**], 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 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[2]. 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 [95]).

Evaluation Versions: To study how examples impact design, we made four

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

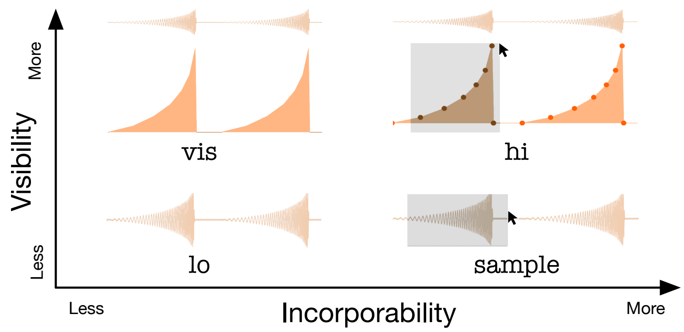


Figure 5.3: Design space for Macaron versions. `hi` and `sample` both allow for selection and copying of example keyframes. `vis` and `hi` both show the underlying profiles. `lo` represents the current status quo; only a waveform is shown.

gallery versions by sampling two theoretical dimensions of example access: element *incorporability* and internal parameter *visibility* (Figure 5.3, 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 [95], piloted them to confirm example variety, then regenerated keyframed versions with Macaron.

5.2 Study Methods

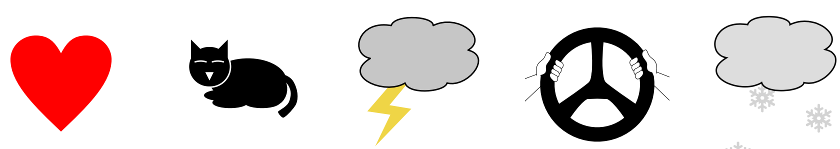
Participants were tasked with creating a sensation to accompany five animations (Figure 5.4) – SVGs (scalable vector graphics) which can be played or scrubbed by the same means as navigating Macaron's time control. We chose animation variety (concrete to abstract) and complexity to inspire non-obvious solutions without 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 5.4'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,

`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.

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).



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

Figure 5.4: 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.

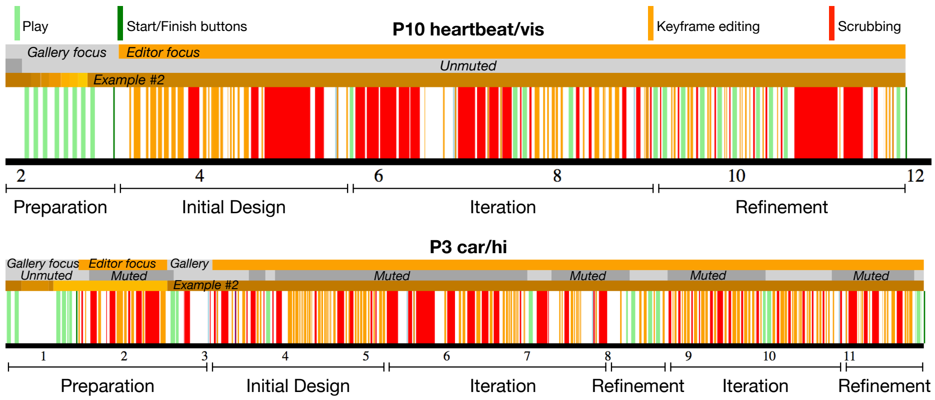


Figure 5.5: Log visualizations showing archetypal design process. Top: P10's heartbeat/`vis` condition (an"ideal" version). Bottom: P3's car/`hi` condition (variations: a return to example browsing after editing, repeated refinement, muted editing).

5.3 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 [14]) and thematic analysis and clustering [77]. We visualized logs using D3 (Figure 5.5). 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.

Prepare	All participants began with a problem preparation step [106]. They played the animation to understand the problem, then typically looked at several (sometimes all) examples. Only P2, P8, and P9 had a task where they did not begin with an example. Otherwise, participants browsed examples, chose a best match to the animation (*"I was trying to find the best match with the visual"* (P7, heartbeat/`hi`)), then transferred into initial design. Participants rarely returned to examples for more exploration; only P3 (car/`hi`) and P5 (car/`lo`) switched to a different example after beginning their initial design. Preparation is characterized by a large number of plays and example switches: on average, 47.45% of all session plays were before the first 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 wasn't close enough to what they wanted to do. Participants typically recreated the example in their editor by copy/paste of the entire design (P1,2,4-8,10) or sometimes a component (P3,10) in incorporable conditions (`hi` and `sample`), or by manually recreating the design (P5,6) or a component (7,10) with `vis`. In the `lo` condition, we only observed P5 somewhat recreating an example. Occasionally, participants would create a new design loosely based on the example rather than recreating it (P3,4,6-8), when using the *Inspire* example use strategy (described later).
Iterate	Participants refined designs with longer periods of editing typically book-ended by playing the entire design (discussed as "real-time feedback" micro interaction pattern). In some cases, especially when the example was "close enough", participants skipped iteration (*Adjust* or *Select* example use strategies, described later).
Refine	Smaller changes forecast design conclusion, e.g., incremental global changes: constant frequency (P1,2,5,6,10), alignment (P1,3,6), or pulse height adjustment (P1,3,8,10). This step is sometimes visible in activity logs, as most participants (P1,3-10) exhibited more frequent plays of the entire design, and shorter periods of editing/scrubbing. Occasionally, participants repeated larger iterations and refinement (P3 car/`hi`, Figure 5.5).

Table 5.2: Steps in observed archetypal design process.

5.3.1 Archetypal Design Process

Log visualizations (Figure 5.5) 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 of the basic process in Table 5.2, to document behaviour and frame discussion.

5.3.2 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 5.6a; P1,2,3,4,7,9): proceed through the timeline, creating amplitude and frequency at the same time.

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

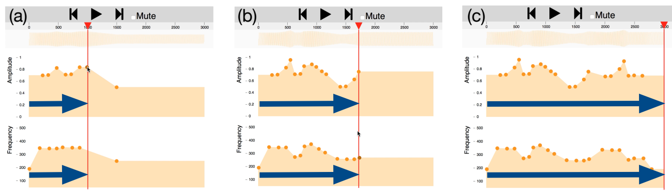
– *Track* (Figure 5.6c, P2,3,6,7,8-10): proceed through one entire *track* (typically amplitude), then the other one.

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).

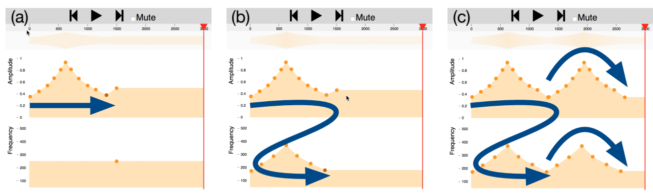
Further showing diverse workflows, participants requested more powerful controls to work with keyframes as a group, such as widen (P5), reverse (P7), shift everything (P9), move up/down and smooth (P4). Other requested features include looping (P1), hovering over a point to see the value (P1), more detail 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 5.6ab) 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 frequency) (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).

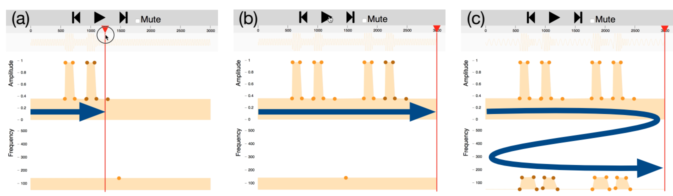
Copy/paste was used for improved work efficiency (especially helpful during initial layout or when creating long or repeating designs) and precision: *"Copy and paste...was also the most precise, because if you feel like it's a perfect fit, you can use it exactly"* (P6). Correspondingly, conditions without copy/paste (*i.e.*, `lo`



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



(b) P6's car/`lo` design progressed by component, developing the component then repeating it.



(c) P10's heartbeat/`vis` design progressed by track. Amplitude was developed first, then frequency.

Figure 5.6: Participants created their designs using different progression paths, suggesting flexibility.

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 5.7. 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;

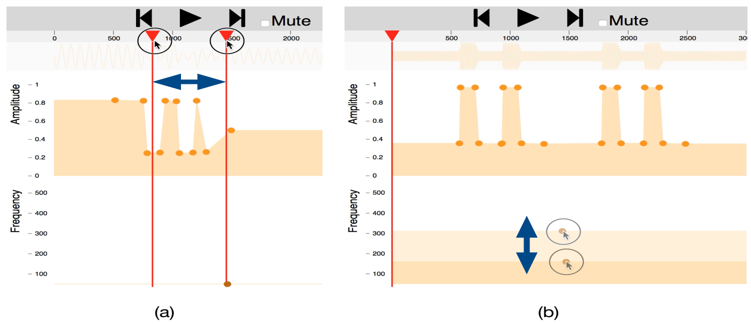


Figure 5.7: Participants used real-time feedback to explore, both (a) in time by scrubbing back and forth (P3 lightning/`lo`), and (b) by moving keyframes (P10 heartbeat/`vis`).

others did not use muting.

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).

5.3.3 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 de-

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.

sign; and to *indirectly scaffold learning* throughout a session. The latter was related to additional themes: task difficulty and individual differences.

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.

5.4 Discussion

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

5.4.1 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 [59] and VT visualization tools like VibViz [95].

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 [24, 106].

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 [44]).

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., [99, 100]).

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.

5.4.2 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. ⌈**OS** *TODO: Comment more on the Online Deployment, probably as its own section, and with image of the demo and current directions. Also need to show MacaronKits.*⌋

Chapter 6

Share: HapTurk

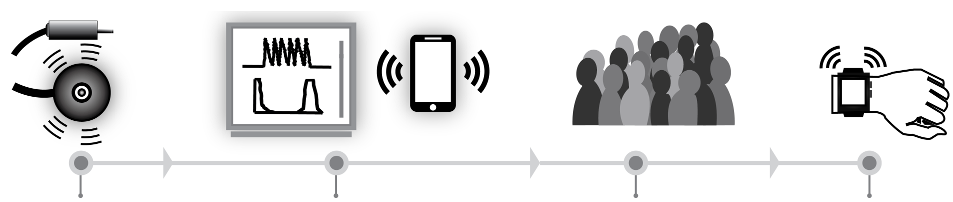
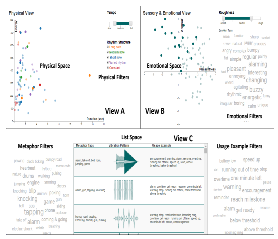


Figure 6.1: In HapTurk, we access large-scale feedback on informational effectiveness of high-fidelity vibrations after translating them into proxies of various modalities, rendering important characteristics in a crowdsource-friendly way.

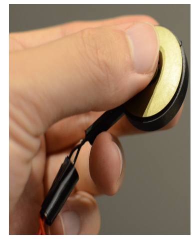
While Chapters 3-5 document iterative development of VT tools, in Chapter 6 we recount a designed *technique* to support large scale feedback high-fidelity VT effects: HapTurk. As we have shown, tactile design relies heavily on iteration and user feedback [91]. In Chapter 3 we found utility in informal, collocated user feedback, but were unable to reach conclusions about haptic language due to small sample size.

In other design domains, crowdsourcing enables collecting feedback at scale. Researchers and designers use platforms like Amazon's Mechanical Turk[1] 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

[1]Also known as "MTurk", `www.mturk.com`



(a) VibViz interface [95]



(b) C2 tactor

Figure 6.2: Source of high-fidelity vibrations and perceptual rating scales.

is unpredictable and uncontrollable. Our approach is to instead send more easily-shared stimuli: *proxies*, which are sent to the crowd instead of the source stimuli. In this chapter, we design and evaluate the potential of two proxy methods for high-fidelity vibrations: visualizations and low-fidelity phone vibrations.

6.1 Approach - Proxy Vibrations

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? In this chapter, we describe and assess two proxy modalities' development, translation process, validation with a test set translation, and MTurk deployment. 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 Design

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 character-

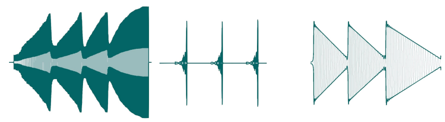


Figure 6.3: VIS$_{\mathrm{DIR}}$ Visualization, based on VibViz

istics, 2) represent the variation in a larger source library, and 3) be small enough for experimental feasibility.

6.2.1 Affective properties and rating scales

To evaluate our proxies, we adapted six rating scales from the tactile literature and new studies. Seifi *et al* [95] 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 [95] 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.

6.2.2 High-fidelity references

We chose 10 vibrations from a large, freely available library of 120 vibrations (VibViz, [95]), browsable through five descriptive taxonomies, and ratings of taxonomic properties. Vibrations were designed for an Engineering Acoustics C2 tactor, a high-fidelity, wearable-suitable voice coil, commonly used in haptic research [95]. 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 6.2b).

6.2.3 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

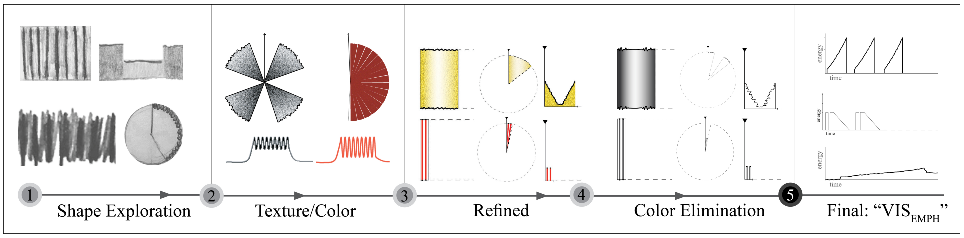


Figure 6.4: Visualization design process. Iterative development and piloting results in the VIS$_{\text{EMPH}}$ visualization pattern.

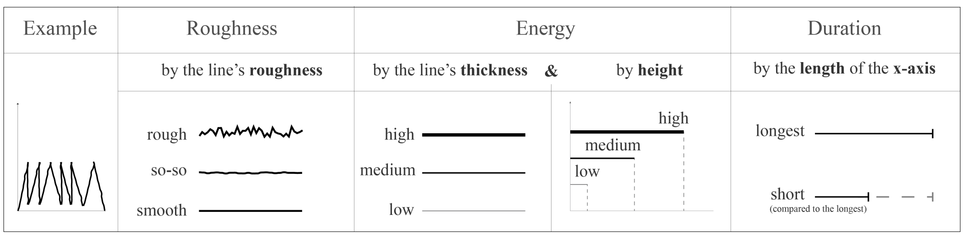


Figure 6.5: Final VIS$_{\text{EMPH}}$ visualization guide, used by researchers to create VIS$_{\text{EMPH}}$ proxy vibrations and provided to participants during VIS$_{\text{EMPH}}$ study conditions.

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 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. [12]) directed [95] 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 [12] to deliberate design-for-lofi [47]), 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.

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

VIS$_{\text{DIR}}$ was based on the original waveform visualization used in VibViz (Figure 6.3). 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 6.4). 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 6.5).

6.2.5 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$_{\text{EMPH}}$, we focused on physical properties when developing LOFIVIB(our single low-fi proxy exemplar). We emphasized rhythm structure, an important design parameter [102] 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 [47], illustrated in Figure 6.6.

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$_{\text{EMPH}}$, we developed LOFIVIB vibrations iteratively, first with team feedback, then informal interviews (n=6).

6.3 Studies

We ran two user studies to evaluate our proxy methods. Study 1 was an in-lab comparison between our high-fidelity reference vibrations and our proxies. Study 2 was an MTurk-deployed comparison of our proxies compared with both in-lab high-fidelity references and in-lab proxies. In both cases we use equivalence testing for statistical analysis.

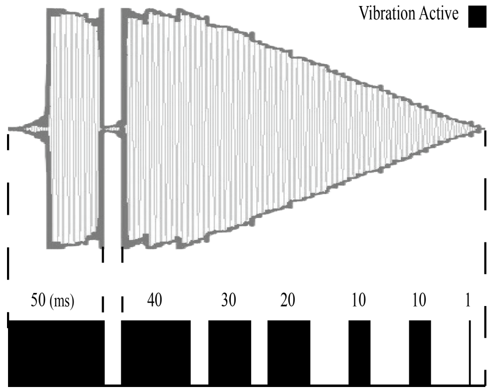


Figure 6.6: Example of LOFIVIB proxy design. Pulse duration was hand-tuned to represent length and intensity, using duty cycle to express dynamics such as ramps and oscillations.

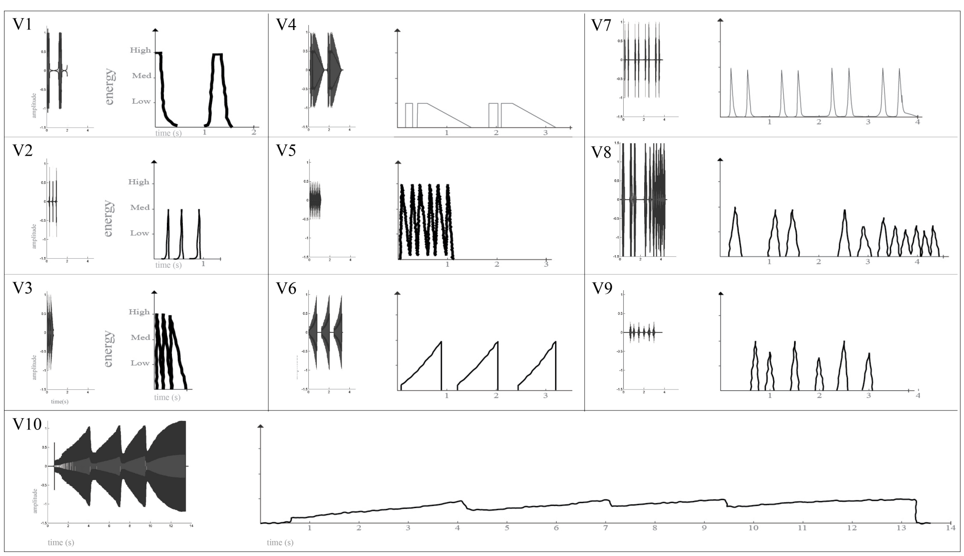


Figure 6.7: Vibrations visualized as both VIS_{DIR} (left of each pair) and VIS_{EMPH}.

6.3.1 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 6.9 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.

6.3.2 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. Participants 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}.

6.3.3 Proxy Validation (Study 1) Results and Discussion

Overview of Results – Study 1 results appear graphically in Figure 6.8. 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 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_{EMPH} [6:1]; VIS_{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 Duration 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_{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_{EMPH} attained only [4:5], and VIS_{DIR} [3:5]. Speed was less translatable. LOFIVIB did best at [4:2]; VIS_{DIR} reached only [1:6], and VIS_{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* $\textbf{VIS}_{\textbf{EMPH}}$** – 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_{EMPH} [4:3], VIS_{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

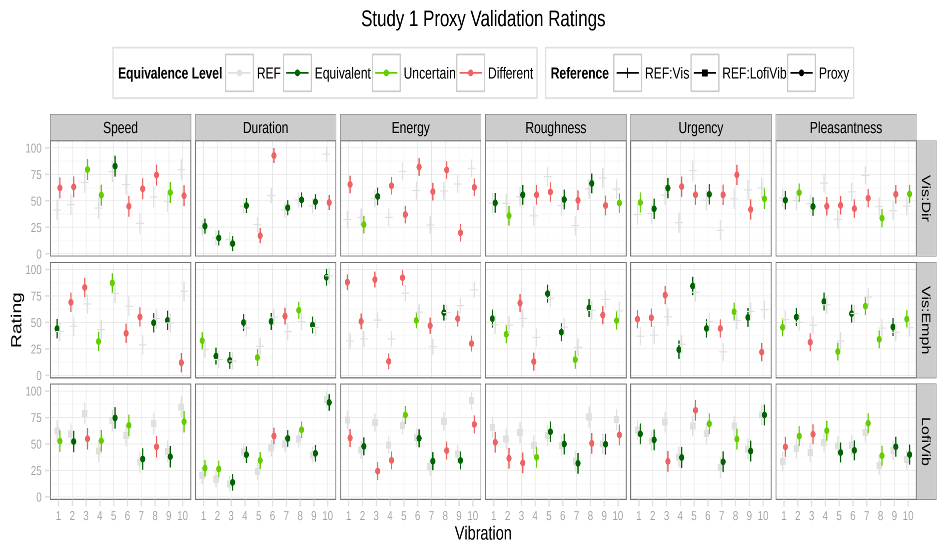


Figure 6.8: 95% confidence intervals and equivalence test results for Study 1 - Proxy Validation. Grey represents REF ratings. Dark green maps equivalence within our defined threshold, and red a statistical difference indicating an introduced bias; light green results are inconclusive. Within each cell, 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.

Energy; VIS$_{\text{EMPH}}$ and VIS$_{\text{DIR}}$ struggled at [1:8].

Emphasized visualization outperformed direct plot – Though it depended on the vibration, VIS$_{\text{EMPH}}$ outperformed VIS$_{\text{DIR}}$ 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$_{\text{DIR}}$ had both more equivalencies and more differences [7:3] versus [6:1] In addition, 16/20 participants (80%) preferred VIS$_{\text{EMPH}}$ to VIS$_{\text{DIR}}$. Although not always clear-cut, these comparisons overall indicate that our VIS$_{\text{EMPH}}$ 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

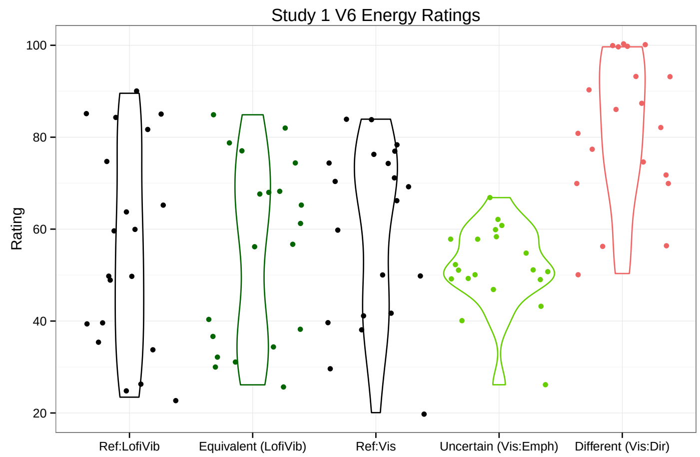


Figure 6.9: Rating distributions from Study 1, using V6 Energy as an example. These violin plots illustrate 1) the large variance in participant ratings, and 2) how equivalence thresholds reflect the data. When equivalent, proxy ratings are visibly similar to REF. When uncertain, ratings follow a distribution with unclear differences. When different, there is a clear shift.

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.

Study 1 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 motivation to access larger samples). Even so, both differences and equivalencies were found in every rating/proxy modality pairing. Most

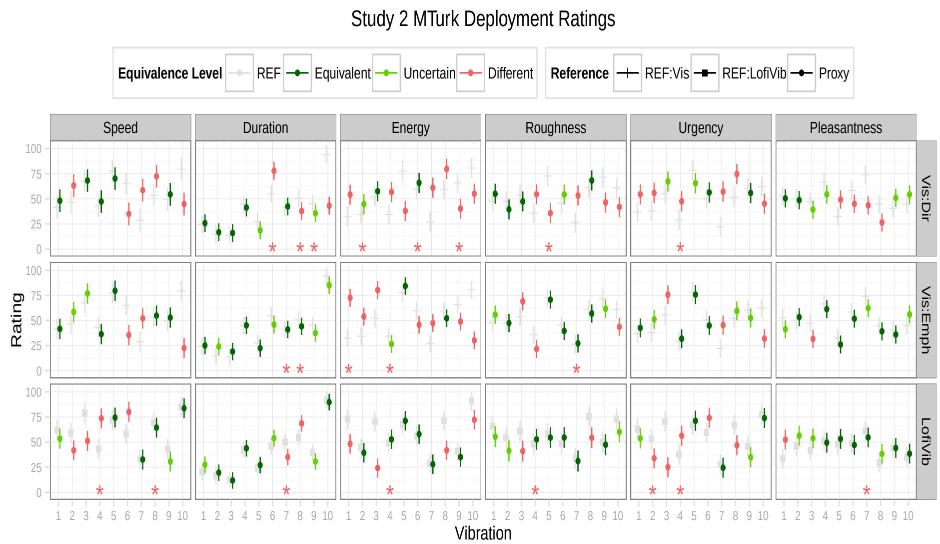


Figure 6.10: 95% Confidence Intervals and Equivalence Test Results for Study 2 - MTurk Deployment Validation. Equivalence is indicated with dark green, difference is indicated with red, and uncertainty with light green. Red star indicates statistically significant difference between remote and local proxy ratings.

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.

6.3.4 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.

6.3.5 Study 2 Results

Study 2 results appear in Figure 6.10, which compares remotely collected ratings with locally collected ratings for the respective reference (the same reference as for Figure 6.8). 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.

6.4 Discussion

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

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.

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.

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.

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.

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.

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 [95], 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.

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).

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.

6.4.1 Limitations

A potential confound was introduced by $\textsc{Vis}_{\textsc{EMPH}}$ having a longer time axis than $\textsc{Vis}_{\textsc{DIR}}$: some of $\textsc{Vis}_{\textsc{EMPH}}$'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. ⌈**OS** *TODO: Connect more with other chapter discussions and intros.*⌋

Chapter 7

Applications

Chapters 3-6 provide rich provide rich but focused data on how to create vibrotactile (VT) experience design tools, and Chapter 7 provides a detailed look at a VT technique for large-scale feedback. To complement these studies, I participated in several more focused projects, to examine other application areas, categories of devices, and gain first-hand experience as a haptic designer:

7.1 **FeelCraft**, a plug-in architecture for distributing customizable feel effects.

7.2 **Feel Messenger**, a design project creating expressive shareable VT icons on commodity smart phones.

7.3 **RoughSketch**, a design project for a drawing application using programmable friction with the TPad phone.

7.4 **HandsOn**, a conceptual model for DIY force-feedback haptics in education.

7.5 **Voodle** and **MacaronBit**, design tools extended to control CuddleBits, simple affective robots.

7.1 FeelCraft

As shown in prior work (cite Hasti), and as we will discuss in Chapter 8, customization is an important feature for haptic experiences. In addition, haptic media must be built around existing infrastructure, as it is not directly supported by most media types. FeelCraft is a media plugin architecture 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 this chapter, we describe the plug-in architecture, envisioned applications, and our implementation for VT grid arrays displaying Feel Effects (FEs) [46] for a popular video game, MineCraft. ⌈**OS** *TODO: Go through citations in this section, update from the paper.*⌋

7.1.1 FeelCraft Plugin and Architecture

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). A pictorial description of the FeelCraft architecture is shown in Fig. 1 architecture.

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.

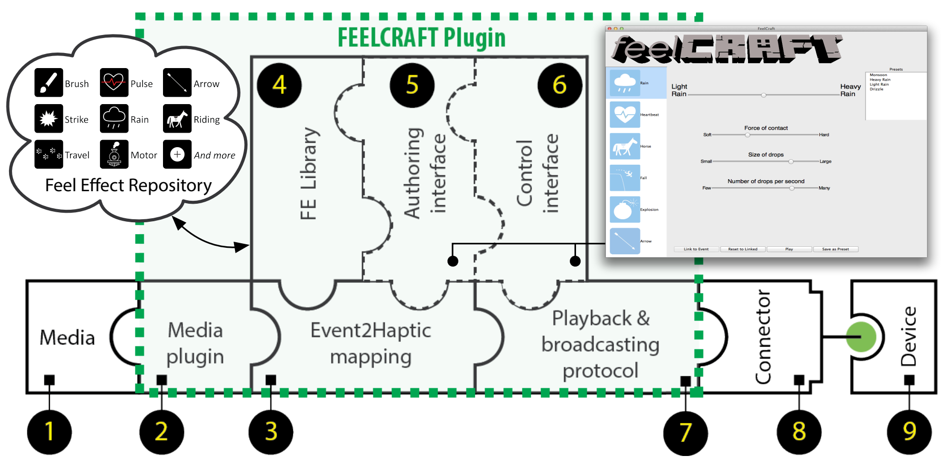


Figure 7.1: FeelCraft architecture. The FeelCraft plugin is highlighted in green. The FE library can connect to shared feel effect repositories to download or upload new FEs. A screenshot of our combined authoring and control interface is on the right.

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.

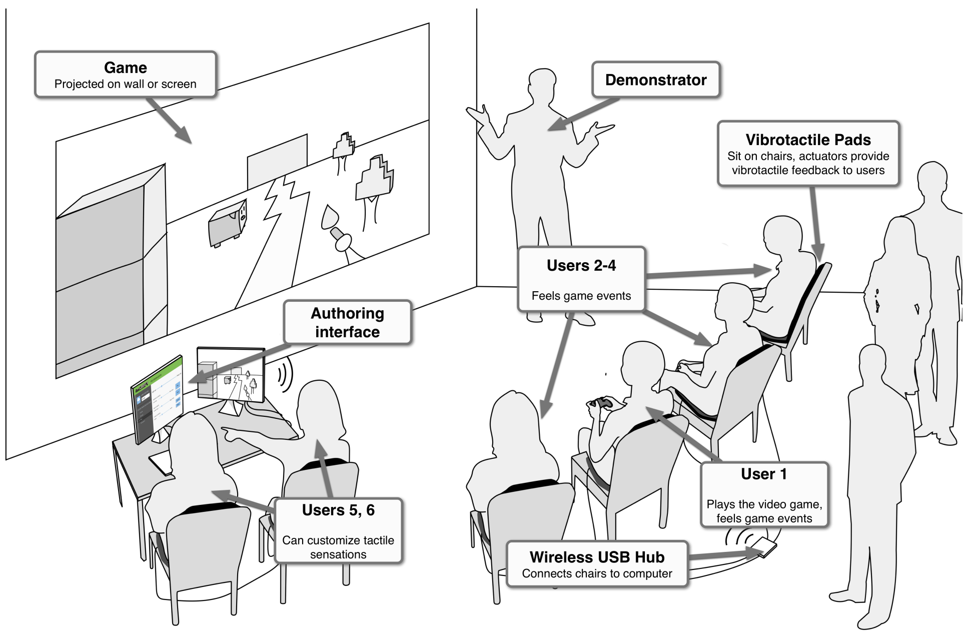


Figure 7.2: Mockup for FeelCraft demo system (cite UIST and AH)

7.1.2 Application Ecosystem

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

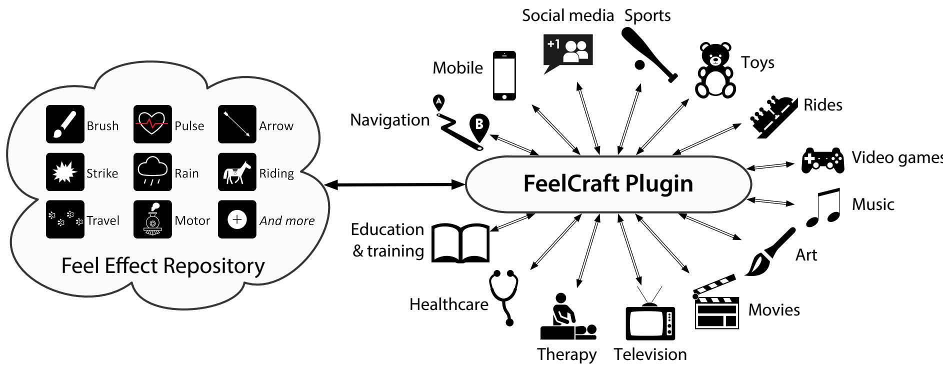


Figure 7.3: Application ecosystem for FeelCraft and an FE repository

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.

7.2 Feel Messenger

In Section 7.1, we designed expressive spatial VT Feel Effects using existing infrastructure via a plugin architecture for desktop applications. In Section 7.2, we look at the expressiveness of existing infrastructure on Android smartphones for customizable VT effects by implementing customizable VT emojis in a chat program, Feel Messenger [47]. As explained previously by Figure 6.6 in Chapter 6, APIs for vibrotactile feedback are limited to a series of pulses. Here we really wanted to push the limit on what was possible with commodity hardware and APIs, and explore how VT effects could fit into current devices. In this project, I play the role of a haptic UX designer.

7.2.1 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

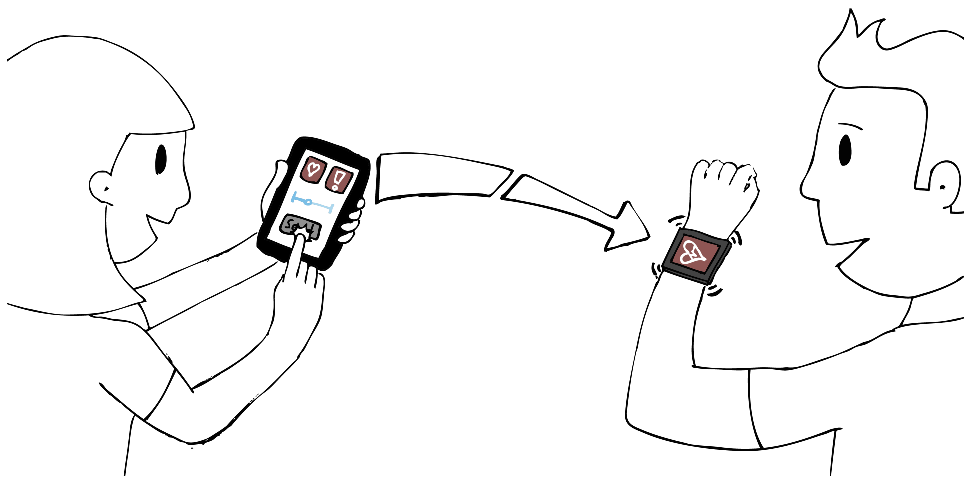


Figure 7.4: Users exchanging expressive haptic messages on consumer embedded devices.

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.

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. By clicking a FE icon, sliders corresponding to parameters (feelbits) are enabled. These sliders may 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.

7.2.2 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.

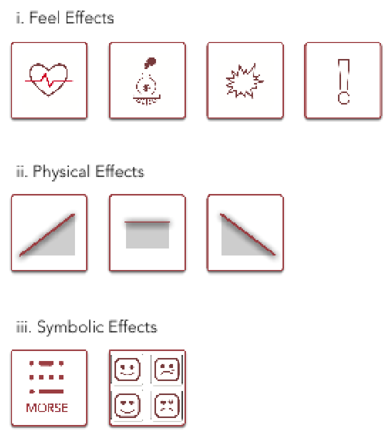


Figure 7.5: Graphical representation of haptic vocabularies and icons.

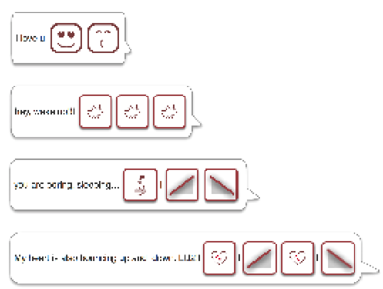


Figure 7.6: Some examples of expressive haptic messages embedded with normal text messages.

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.

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.

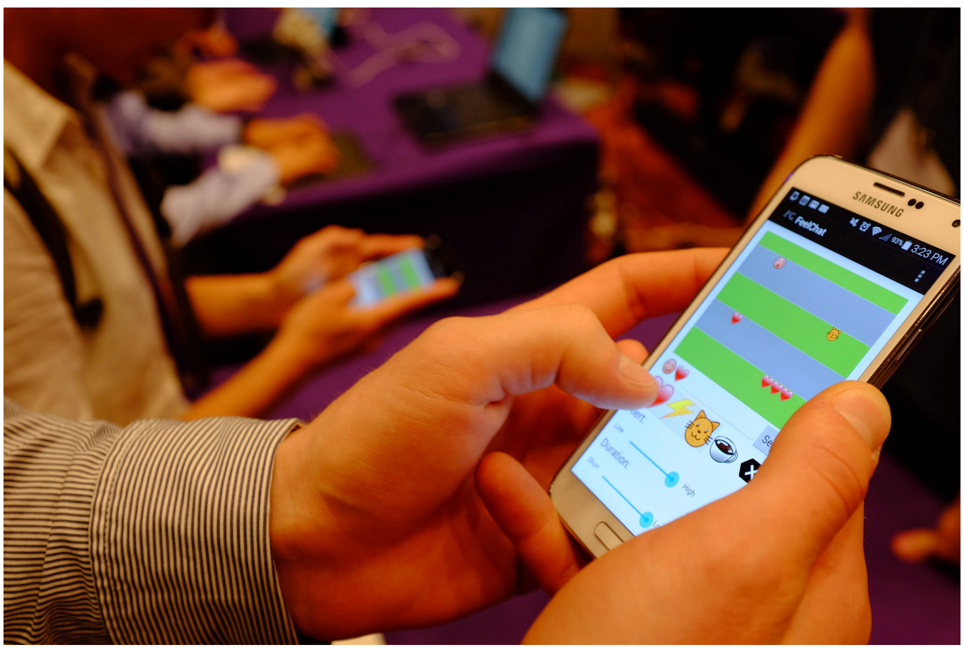


Figure 7.7: Implemented Feel Messenger application demo at World Haptics 2015 (cite).

7.2.3 Demo

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

In this prototype, we explored both predefined effects and Type 1 icons (Feel Effects). Our predefined effects were designed with 6 emoji. Our four Type 1 FEs were: Heartbeat (Pulse FE), Lightning (Strike FE), Cat Purr (Motor FE), and Coffee (Urgency FE). These VT emoji could be embedded in chat messages, sent between two Android phones using UDP. VT effects are felt when editing, when received, and when the user taps a message. All effects were implemented using built-in Android APIs.

7.3 RoughSketch

FeelCraft (Section 7.1) and Feel Messenger (Section 7.2) used existing infrastructure to conduct VT design, but how do other types of feedback vary? In Sections 7.3 to 7.5, we investigate other modalities in other applications. Here, in Section 7.3, we describe RoughSketch, a drawing application for the TPad Phone.

The TPad Phone (www.thetpadphone.com) is a programmable friction display mounted on an Android phone. It uses piezo-actuated mechanical vibration to create a ⌈**OS** *TODO: need actual name for the effect, and probably to cite original TPad papers*⌋. As part of the World Haptics 2015 Student Innovation Challenge, we built RoughSketch, a mobile drawing application to explore friction displays for digital mark-making. We looked at several mark-making interaction techniques including:

- *Paintbrush*, where you feel paint leaving your finger,
- *Pen and eraser*, based on real-world writing utensils,
- *Spray paint*, where you feel the roughness of paint on the screen as you spray,
- *Pinch/zoom*, inspired by compressing and stretching rubber, and
- *Feel finger*, the ability to feel your drawing on the paper.

To implement RoughSketch, we adapted an open-source Android application (todo) and used the TPad Phone API to control friction using two methods: preloaded textures defined by bitmaps, and code-based envelopes that programmatically adapt friction based on input values or time. We used a variety of real-world metaphors to inspire our designs, and rapidly iterated on them. Illustrations of these are available in Figure 7.8. While designing and developing RoughSketch, we exposed a design space, finding conflicts for our metaphors, specifically, should TPad sensations feel like their real-world equivalent, or are they unique to

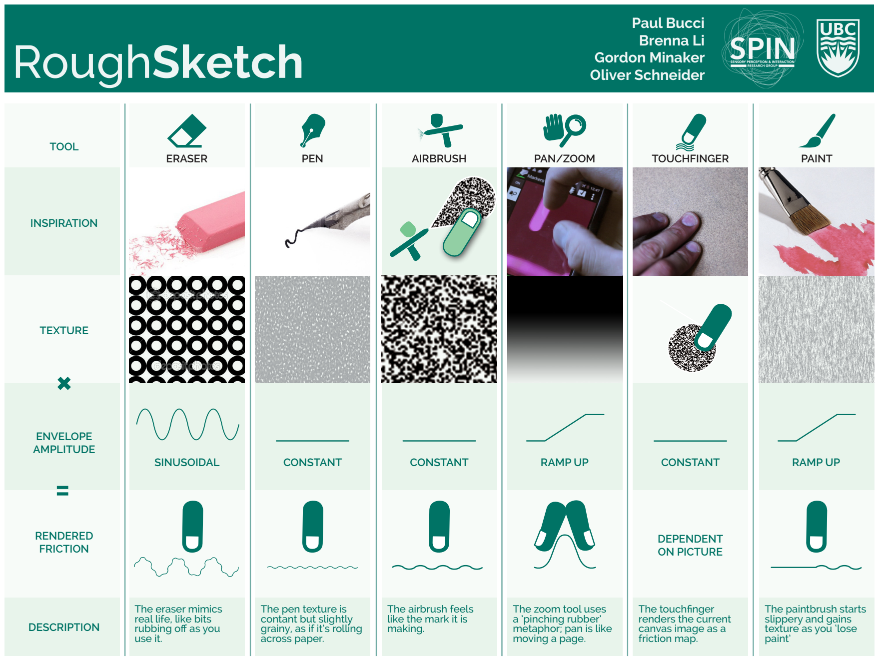


Figure 7.8: RoughSketch handout, illustrating interaction techniques an textures.

the TPad; and should rendered textures represent the drawing process, or the finished product? We also explored how they would be adapted if the user interacted with RoughSketch using an app rather than their finger. These findings are outlined in Figure 7.9.
⌈**OS** *TODO: here or in final discussion, put findings like, some styli worked, and others didn't; and that turning off the tactile sensation was important for convincing people that haptics was valuable, especially when it was subtle; maybe reflect on pinch/zoom?* ⌋

7.4 HandsOn

In Section 7.4, we investigate creative control of 1-degree of freedom (DOF) force-feedback display for education. Education literature has established that embodied, physical interaction can improve learning by making abstractions concrete [**? ? ?**]. Meanwhile, online courses and interactive lesson plans have increased education

Figure 7.9: RoughSketch poster, describing interaction techniques and high-level findings.

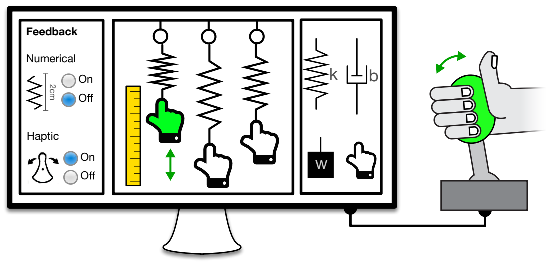


Figure 7.10: Students, teachers, and researchers can explore science, technology, engineering, and math (STEM) abstractions through low-fidelity haptics, incorporating elements into system designs.

access and versatility. Haptic technology can integrate these benefits, especially with recent developments for affordable DIY hardware (cite hapkit), but requires accessible software that enables students to creatively explore haptic environments without writing code.

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.

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 7.11).

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 7.11A) (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 7.11B). 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 7.11C). 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:

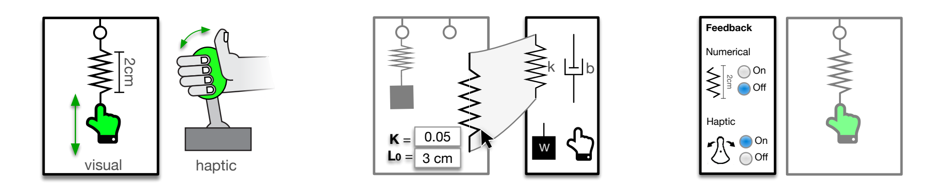
Our first *HandsOn* interface is *SpringSim* (Figure 7.12), which supports a spring lesson – spring systems are natural as a virtual environment of easily-controlled complexity. In *SpringSim*, *objects* include single springs and parallel spring systems, with properties spring rest length (cm), stiffness (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 7.13). 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 7.11: The *HandsOn* CM enables three kinds of exploration based on requirements.

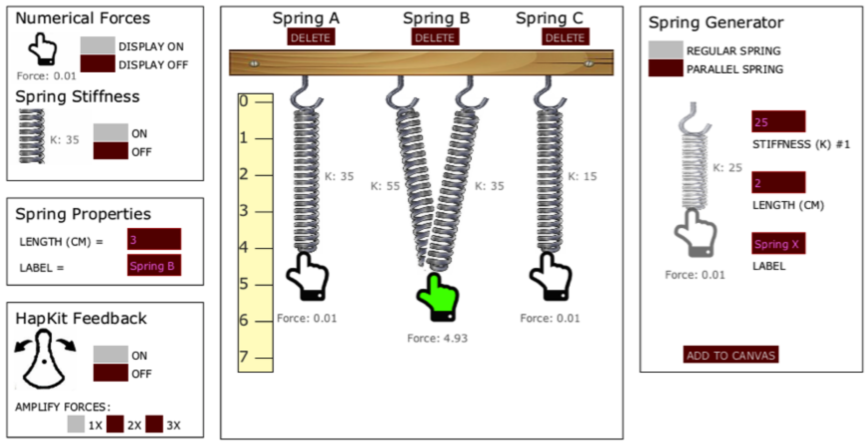


Figure 7.12: *SpringSim* interface, a *HandsOn* sandbox for a single lesson module on springs.

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 ($F(26,336) = 2.10$, $p = 0.00165$).

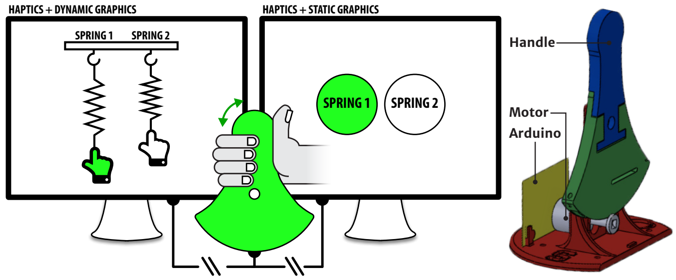


Figure 7.13: In the *Hapkit+Dynamic Graphics* condition, graphical springs responded to input (left); static images were rendered in the *Hapkit+Static Graphics* condition (right); in both, HapKit 3.0[**?**] was used as an input/force-feedback device (far right).

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 [14]. 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.

7.5 CuddleBit

TODO

Chapter 8

Ground: Lessons from Hapticians

High-fidelity haptic technology is expanding user experience in wearables, automobiles, and new media. Haptic experiences have shown promise in multiple interaction aspects, including emotional therapy [**?** **?**], education [**?**], and entertainment [90]. Technological advances are making sensations more compelling at the level of consumer products, making it possible to render variable friction on direct-touch surfaces [64**?**], and forces without grounding to a table or wall [**?** **?**]. Even commodity vibrotactile displays are increasing in expressivity, with high quality actuation and painstaking design a priority in devices like the Apple Watch (www.apple.com) and Pebble (www.pebble.com).

Obstacles to Design – However, content remains scarce and design knowledge is limited. The academic literature suggests many challenges to design, including latency [**?**], individual differences in both programmed [64] and natural [**?**] textures, low-level perceptual variation [**?**], the effect of aging [**?** **?**], and interpretation of effects [95**?**], often due to personal experience [91]. These are further reinforced by personal requests to the authors from industry practitioners. We expect there to be many challenges facing industry designers, but have little direct evidence to reinforce this and guide research. Part of this comes from the rareness of professionals designing haptics explicitly, and absence of studies of their workflow. In this work, we take a first in-depth look at haptic designers' experiences to confirm, correct, and elaborate expected challenges, and connect HaXD to other fields of design.

Haptic Experience Design – We define 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. Much of what we find could also be gainfully applied to the design of tangibles, even with their lack of actuation, although we leave them out of our scope to focus on actuated interfaces. For brevity, we use "haptic designers" to refer to haptic *experience* designers.

Target Audience – We expect haptic experience designers will find our results familiar, and hope the articulated challenges and recommendations are useful for their practice. However, we primarily target people who are one step removed from HaXD, e.g., , with other design, haptics, or business expertise.

We expect **non-haptic *design* experts** will find the specific challenges to HaXD informative by revealing processes of design that are invisible or taken-for-granted in other fields. We also hope designers might lend their expertise to accelerate creativity tools for these complex physical interactive systems.

We expect **non-design *haptic* experts** might further develop an appreciation for how design work fits into haptic technology, and understand how their devices and research findings are applied in practice. Our recommendations also motivate several avenues for both basic and applied haptic research.

Finally, we expect **industry partners** will gain insight in how the business world interfaces with haptic technology. This includes those already involved with haptics or similar technologies like wearables, and those looking to become involved. We hope our findings will cultivate connections between the diverse stakeholders interacting with HaXD, and that our challenges (and thus opportunities) might inspire people to work more with this emerging modality.

RoadMap – In this work, we aim to shine a light into the dark land of HaXD. In **Section 8.2**, we report a grounded theory [14] analysis of interviews with six professional haptic designers at four levels: the goals of and processes used by haptic designers; the complex, vertical natures of the experiences they design; the collaborative ecosystem in which our designers work play multiple roles; and the influence of the cultural context of haptics, its value, and risk. In **Section 8.3**, we follow up on the themes on tools and collaboration discovered in these interviews with a broader sector of industry and academic designers, accessed through a quantitative survey and qualitative suggestions from a workshop at a major haptics conference (World Haptics 2015). Finally, in **??**, we synthesize and discuss our findings in three major groups:

1. A description of HaXD activities showing it is already a subfield of design.
2. A list of challenges facing haptic designers, showing HaXD requires unique considerations when compared to other, more established fields.
3. A set of recommendations for driving HaXD to become more established.

We conclude by imagining what a mature discipline of HaXD might look like.

8.1 Related Work

In this section, we cover design thinking and theory, then review haptic technology and perception, and conclude with previous efforts to understand or support HaXD.

8.1.1 Design Thinking

Design thinking has become an empowering way to approach technology and user experiences, focused on rapidly generating, evaluating, and iterating on multiple ideas. In this work, we adopt an embodied perspective, viewing cognition, and thus design, as an activity that occurs both in physical and social worlds [42**?**]. This perspective is particularly appropriate for HaXD as it matches recent thinking for interaction design [**?**] and suits the physical, embodied nature of haptic technology. We organize this section into three major areas involving different aspects of the designer's context: pre-design problem preparation, hands-on activities like sketching, and effect of collaboration.

Problem Preparation – Also known as the "problem setting" [93], "analysis of problem" [106], or "collect" [98] step, problem preparation involves getting a handle on the problem and drawing inspiration from previous work. Creative acts are more accurately seen as recombination of existing ideas, with a twist of novelty or spark of innovation by the individual creator [106]. To create something new, a creative act requires an understanding of the domain (symbolic language of the field) [**?**]. Schön demonstrated that designers initially frame their problems before developing a solution in order to match it to their repertoire, their collected professional (and personal) experience [93]. External examples are especially useful for inspiration and aiding initial design [10, 40]; early and repeated exposure can increase creativity, but late exposure carries a risk of conformity [**?**]. We show evidence that haptic designers follow these processes, with a dedicated problem preparation step and various collections of examples.

Hands-On Design – Experience design is often described as a funnel of idea candidates, where the designer iteratively generates and refines multiple ideas in parallel until a final, developed design (or set of designs) remains [10]. Developing ideas in parallel can facilitate generation and evaluation, delaying commitment to a single design [36, 86], while in groups, sharing multiple designs improves variety and quality [24].

Sketching supports ideation, evaluation, and multiple ideas, allowing the designer to explicitly make moves in a game-like conversation with the problem [93]. Some researchers even declare sketching to be the fundamental language of design, like mathematics is the language for scientific thinking [16]. Sketching is rapid and exploits ambiguity, allowing partial views 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. To explore design spaces creatively, designers must be able to rapidly undo, copy and paste, and see a history of progress; they need"high ceiling" and "wide walls" [86]. We are not the first to suggest sketching for haptic technology [75, 76], though we find that this fluidity is hampered by painful iteration cycles.

Collaboration – Design is a collaborative process; involving more people increases the potential for generating more varied ideas [106], and is important for creativity support tools [86, 98]. Although sometimes group dynamics influence the design process negatively, proper group management and sharing of multiple ideas results in more creativity and better designs [40], and can even influence the work of crowds [**?**]. Collaboration takes many forms, including informal *relating* with colleagues, and *donating* information to the public/annals of time [98]. Csikszentmihalyi makes the provocative statement that creative acts, to be creative, must be accepted as useful by members of a field [**?**]. Collaboration can also be organized into two dimensions: collocated (same location) or distributed (different locations), and synchronous (simultaneous) or asynchronous (at different times) [27]. As far as we are aware, we present the first description of the HaXD's collaborative ecosystem in Section 8.2.3; many of these contexts require extra consideration for haptic technology.

8.1.2 Haptic Perception and 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. Here, we focus on technologies most relevant to those discussed by our participants. More detailed introductions can be found in elsewhere [39**?**].

Tactile Perception and Technology – Tactile sensations rely on multiple sensory organs in the skin, each of which detect different properties, e.g., , Merkel disks detect pressure or fine details, Meissner corpuscles detect fast, light sensations (flutter), Ruffini endings detect stretch, and Pacinian corpuscles detect vibration [**?**]. Of these, the Pacinian corpuscle is most targeted by technology through vibrotactile (VT) sensations, where vibrations stimulate the skin. The most common example of this is smartphone vibrations. VT actuators can take may forms. Eccentric mass motors (sometimes "rumble motors" or, when small, "pager motors") are found in many mobile devices and game controllers, and are affordable but inexpressive. More expressive are tactors, or voice coils, implemented in a variety of ways and offering two degrees of freedom: frequency and amplitude. Piezo actuation is a very responsive technique that is typically more expensive. While tactors directly stimulate the skin, linear resonant actuators (LRAs) shake a mass back and forth to vibrate a handset in an expressive way; a common research example is the Haptuator [110]. Instead of directly stimulating the skin, this actuator typically shakes another device held by the user, such as a mobile device [111] or pen [20]. As of writing (2016), this type of actuator is increasingly deployed in mobile contexts (e.g., , the Apple Watch Taptic engine).

Proprioception and Force Feedback – Proprioception, the sense of force and position, is synthesized from multiple sensors as well: the muscle spindle (embedded in muscles), golgi-tendon organ (GTO) in tendons, and tactile and visual cues [52]. 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. At other times, entire screens might push back on the user in a single degree-of-freedom.

8.1.3 Previous efforts for HaXD

Several software, hardware, and conceptual tools have been developed to help haptic designers.

Software – Software support for haptic technology tends to manifest in three ways: large collections of haptic sensations, programming toolkits, and authoring tools.

Sensation collections typically support VT stimuli. The UPenn Texture Toolkit contains 100 texture models created from recorded data, rendered through VT actuators and impedance-type force feedback devices [20]. The Feel Effect library [46], implemented in FeelCraft [90], lets users control sensations using semantic parameters, "heartbeat intensity" respectively. Immersion's Haptic SDK (immersion.com) connect to mobile applications, augmenting Android's native vibration library. VibViz [95] offers 120 vibrations organized around many different perceptual facets.

Force-feedback environments tend to be supported through programming toolkits. CHAI3D (chai3d.org), H3D (h3dapi.org), and OpenHaptics (geomagic.com) are major efforts to simplify force-rendering. Table-top haptic pucks can use the HapticTouch Toolkit [58], which includes parametric adjustment (e.g., , "softness") and programming support.

The haptics community has long recognized a need for non-programming compositional tools [35]; these tend to focus on VT stimuli or simple 1-degree-of-freedom force feedback. Many editors [28, 89, 99**? ?** , 100] use graphical mathematical representations to edit either waveforms or profiles of dynamic parameters (torque, frequency, friction) over time. Of these, Vivitouch Studio [100] offers the most integration with other modalities in games, and Macaron [**?**] offers most accessibility by being online. The Vibrotactile Score [62] uses a musical metaphor, and is preferable to programming, but requires musical experience [60]. Mobile "sketching" tools like the Demonstration-Based Editor [41] and mHIVE, a Haptic Instrument [91] are useful for exploration, but not refinement. Since iOS 5 (2011), Apple has let end-users create on/off vibrations as custom vibration ringtones. Immersion's Touch Effects Studio lets users enhance a video from a library of tactile icons supplied on a mobile platform. Actuator sequencing [84], movie editing [54], and animation [**?**] metaphors enable multi-actuator, spatio-temporal VT editing. Though many of these tools are founded in an understanding of haptic designers' needs [100**?**] or begin to capture a slice of the HaXD process [**?**], they do not fully capture the context and activities of haptic designers.

Hardware – 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 [34]. More recently, Wooden Haptics gives open-source access to fast laser cutting techniques for force feedback development [31].These platforms, especially Arduino, have made significant improvements to enable rapid iteration and hardware sketching.

Conceptual tools – Some higher-level perspectives offer outcome targets or design attitudes to guide haptic practitioners. "DIY Haptics" categorize feedback styles and design principles [39, 68]. "Ambience" is proposed as one target for a haptic experience [69]. Haptic illusions can serve as concise ways to explore the sense of touch, explain concepts to novices and inspire interfaces [37]. "Simple Haptics", epitomized by *haptic sketching*, emphasizes rapid, hands-on exploration of a creative space [75, 76]. Haptic Cinematography [23] uses a film-making lens, discussing physical effects using cinematographic concepts. Haptics courses are taught with a variety of foci including perception, control, and design, providing students with an initial repertoire of skills [50, 83].

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 [28, 80]. Language is a promising way of capturing user experience [79], and can reveal useful parameters, e.g., how pressure influences affect [112]. Tools for customization by end-users, rather than expert designers, are another way to understand perceptual dimensions [94, 95]. However, this work is far from complete; touch is difficult to describe, and some even question the existence of a tactile language [48].

8.2 Interview Analysis: Haptic Experience Designers in the Wild

In this section, we present the findings from a qualitative analysis of six interviews with professional haptic designers. One researcher analyzed these results with grounded theory [14], with influence from phenomenology [77]. We group our findings with four subsections, found by looking at different levels of context [14] to cluster themes:

- *8.2.1 Designers and process:* a description of our designers and outline of their process;
- *8.2.2 Structure of haptic experiences:* themes about complexity, multimodal nature, the necessarily holistic scope of design modifications, and resulting strategies and challenges;
- *8.2.3 Collaboration:* the different stakeholders, roles of haptic experience design, and related themes; and
- *8.2.4 Cultural context:* value, risk, and pragmatics of haptics affecting designers, users, and customers.

8.2.1 Haptic Experience Designers and Interview Process

One researcher interviewed 6 participants (5 men) in April-May 2012. Each interview lasted between 30-60 minutes and consisted of initial ice-breaker and general open-ended questions, then increasingly detailed questions about participants' processes. Interviews followed a semi-formal, structured qualitative inquiry process that allowed for followup. P2-5 were fully recorded and transcribed, while P1,6 results were collected from interviewer notes alone. As such, results from P2-5 will be reported as quotations, while P1,6 comments be reported as notes. Table 8.1 describes our participants and their projects.

To contextualize this interview data in time, we note that in 2012 the following situation was generally true of professional haptic design. Our impression based on recent interactions with designers in comparable roles in industry is that at the time of this writing (2016) the situation has changed primarily in three respects: (1) hardware has improved and become more diverse, and (2) haptic feedback is more prevalent and expected in the consumer culture. Finally, (3) in a limited number of subdisciplines, e.g., , gaming environments, specialized tools have begun to appear to solve very specific designer problems where the cost/benefit equation merits their development. However, the design pressures shaping practices and tools themselves have changed little, as confirmed by Section 8.3. The themes we present were emergent, discovered while using qualitative techniques of clustering and affinity diagrams [77], memoing, and iterative coding [14].

Design Process Overview – Most of our emergent themes applied to various points throughout the design process. However, we did find that participants followed a familiar process of understanding the design problem, iteratively developing ideas and evaluating them. Here we briefly outline this process, with links for connecting themes.

Project preparation: Participants described the initial stages of a project as a time to establish and understand requirements, gather initial design concepts, and define or negotiate project parameters. Designers often collected examples of haptics for inspiration (Section 8.2.3/Co5), and gathered requirements, both direct requirements for haptic designs (Section 8.2.4/CC1) and project parameters around the value, cost, and risk of haptic technology (Section 8.2.4/CC4,CC5).

Iteratively developing concepts: P2-6 referred an iterative process of some sort; they found different ways to fit it into their collaborative ecosystem and constraints. Often, initial requirements were not actually what clients wanted, and our designers would have to iterate (Section 8.2.4/CC1). P5's teams explicitly follow a conventional user-centered design process, iterating on prototypes and understanding of customer needs. P3 sometimes has to ship mockups and devices back and forth with their customers (Section 8.2.3/Co5). Each design problem faced by our participants had to be treated as a unique problem, with designers fine-tuning their design to fit the problem (Section 8.2.2/Ex5). Our designers used a variety of evaluation techniques to choose their final designs (Section 8.2.4/CC2).

We now proceed with our main cross-cutting themes, organized into three levels of context: the haptic experience and its implementation (Section 8.2.2), the designers' collaborative ecosystem (Section 8.2.3), and implications from the wider cultural context of haptic technology and business requirements (Section 8.2.4). These are outlined in Figure 8.1.

P1 (M, over 15 years of human factors experience, PhD) held a design + human factors position at major healthcare company. He worked with auditory alarms, signals, and emotional experience. Despite a focus on audio, he frequently related his work to haptics and works with physical controls, controlling characteristics like force profiles and detents, and described the haptic and audio processes as being the same. Working in health care means there are tight regulations that need to be followed, and a noisy, diverse environment. P1 used a number of psychology and human factors techniques, such as semantic differential scales, factor analysis, and capturing meaning.
P2 (M, age/experience tbd, apprx. 5-6 yrs in haptics, PhD) described two projects: his experience adding mechanical feedback to touch screens at a major automotive company, and his PhD work on remote tactile feedback, where feedback was displayed on one hand while the other interacted with a touch screen. P2's main concern is "rich feedback", communicating information like affordances to the user. This is both pragmatic, such as "consequences" of the button, and affective, aiming to have sensations "feel right." P2 focused on button presses on a touchscreen, rather than exploring "roughness" of a touchscreen or other surface.
P3 (M, 10 years leadership experience with actuation, sensing, and multimedia, M.Eng.) worked at a company that sells actuators used to add haptics to technology (like a tablet computer, game controller, or mobile phone). P3 had 20-30 projects going on at any time, each with their own level of size, goals, constraints, and other contexts. His main goal was to sell a developed actuator (with several variants).
P4 (M, 11 years of design, development, and analysis/simulation experience, PhD) also puts actuators into new form factors (e.g., touch screens in cars). When he worked, he had limited time and resources, so there is not much time to change things.
P5 (M, 12 years of haptics UX experience, M.Sc.) held a user experience leadership position at major haptics company that sells haptic control technology and content; he described mostly software solutions. His company worked with different domains, but most examples are from mobile phones (handhelds), with a brief mention of automotive haptic feedback. They worked with extremely high-end piezo vibration actuators, high bandwidth (frequency and mechanical), and delivered software solutions for Android to their customers: OEMs (original equipment manufacturers). He described handheld feedback as two different classes: confirmation haptics, like a vibration to indicate a widget has been used, and animations/gestural feedback, which is more complicated.
P6 (F, 5-6 years in haptics, PhD) worked at a major car manufacturing company. She primarily designs "feel" properties such as friction, inertia, and detents of physical controls inside automobiles. P6 also works on active haptic controls. Design aspects include measuring force vs displacement profiles and maintenance of a large scale haptic design specification repository that spans user and technical requirements. This haptic specification repository is used by many engineering and business stakeholders across many sites in different countries.

Table 8.1: Our expert haptic designer participants and their projects. Experience and position are reported from interview year (2012).

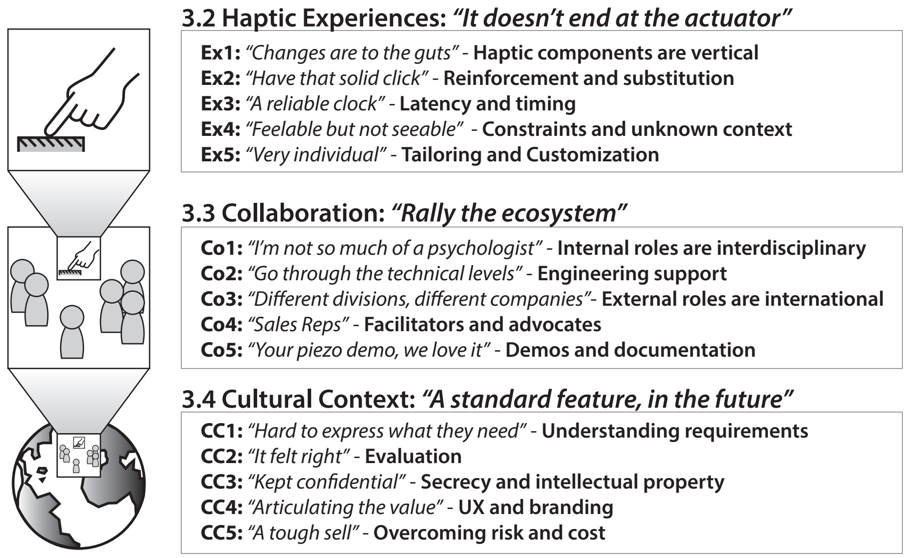


Figure 8.1: Our three super-themes and their sub-themes.

8.2.2 Haptic Experiences: "It doesn't end at the actuator"

As we've discussed, the haptic sense is really a *collection* of senses, synthesizing different material properties together in a holistic way. Visual and audio feedback, grip, materials, and other dynamics all play a part in the final percept.

> *"The problem is it doesn't end at the actuator, there's a lot to do with the case of the device, the mass of the device, the mechanical coupling between the device and the hand...this all comes into play because it's a tangible experience, and so if there's mechanical resonances that get stimulated by the actuator that make it sound noisy, then it becomes a cheap experience, even if it has a piezo actuator."* (P5)

The holistic, context-sensitive nature of haptic experiences has implications for HaXD. Designers both face multimodal constraints and hvae opportunities to circumvent those constraints. We begin by discussing implications for implementation, where haptic components are directly related to the "guts" of the system (Ex1). Then, the opportunities for improving design: reinforcement and substitution are powerful tools for haptic designers (Ex2), as long as they have synchronized with a reliable clock (Ex3). However, multimodal contexts are sometimes uncontrollable or unknown, barring designers from using their tricks (Ex4). We

Table 8.2: Sub-theme summaries for Haptic Experiences super-theme.

Ex1	*"Changes are to the guts"*	Changing a haptic component may influence the larger hardware/software system, and vice-versa.
Ex2	*"Have that solid click"*	Haptic designers can work around constraints through multimodal tricks.
Ex3	*"A reliable clock"*	Without fast feedback and synchronized timing, haptic experiences fall apart.
Ex4	*"Feelable but not seeable"*	Other modalities may impose constraints; constraints may not always be knowable.
Ex5	*"Very individual"*	Designers tailor their solutions to each application; end-users benefit from customization.

finish this section by discussing how haptic experiences are often bespoke, tailored to constraints of known contexts, or customizable to unknown contexts (Ex5).

Ex1: *"Changes are to the guts"* – Haptic components are vertical. Haptic experiences are created the actuating component interacting with most other system components. Changing a haptic component can thus affect the entire system's design, unlike other upgrades, like improving memory in a mobile phone: *"you get the impression every other month they have a new phone...but the guts of it do not change much"* (P3). New phones often just have a faster CPU or more memory swapped into the same system. Designers need to take the system into consideration when adding haptics:

> *"First we had to get the outer dimensions roughly about right, to get the visual impression close to what it resembles later in the application"* (P4).

This effect is bidirectional. Changing the size or material of the casing can have a profound effect on the sensation; correspondingly, any changes to the haptics will have an effect on the entire structure of the device. Changes to software are also cross-cutting: *"we're digging into the source code of Android...we need to make sure that we have the right hooks in the right locations...that's a software architecture issue, right?"* (P5).

Ex2: *"Have that solid click"* – Reinforcement and substitution. When the context is known, designers can use physical system properties to improve haptics: *metal makes unwanted sound, so change it with a plastic* (P6). Knowing the context, like a car dashboard, lets the designer make haptics more convincing with reinforcement,e.g., , adding visual or audio feedback: *"need to have that solid* *[haptic] click at 150 [Hz] plus some audio at 300 or 400 Hz, which is going to give you that sense of quality, and, consistency across the whole dashboard"* (P5).

When a known context has constraints, designers also use substitution to provide haptics. P2 describes two such occasions, one for sensing input and one for displaying output. Because P2 could not sense input pressure, he instead used how long the user was pressing the screen ("dwell time"): *"we were substituting the forces that are needed on the actual buttons with dynamic dwell times"* (P2). In another case, P2 could not actuate a touch screen, so he used tactile feedback on the other hand. With careful timing, the user perceives a single sensation, letting P2 actuate another location, e.g., , the steering wheel: *"somehow you connect these two things, the action with the dominant hand and the reaction that is happening somewhere else"* (P2).

Ex3: *"A reliable clock"* – Latency and Timing. To effectively use reinforcement, and to have effective haptic feedback, timing must be tightly controlled. This is well established in the literature [**?**], and well known to our participants: *"I think, audio feedback and tactile feedback and visual feedback has to happen at a certain time to have a real effect"* (P2). Haptic actuators themselves have become very fast and responsive. Previously, it had been a goal for our designers: *"[what we] strived in the past significantly to do was to push the market towards high mechanical bandwidth actuators, so actuators that can respond in 15 milliseconds or less"* (P5). Now, high-quality actuators are the main competitive advantage our designers offer:

> *"High-definition feels over a very broad frequency range, with enough strength and small enough, and especially very fast response time, that's our business"* (P3).

Designers now focus on introducing no new delays to the stack. P2 describes unintentionally adding latency to one project: *"we had this Python program and Arduino and all this communication going on"*. P2 *"threw out some of the serial communication which [had] made the whole thing a little slow"*, and thus, the *"latency again felt right"*. Timing problems between components can happen at any time: *"we've gotten in situations before where we've been very near to completion in design projects, and for whatever reason we can't get a reliable clock, from the CPU, then the whole thing falls apart"* (P5).

Ex4: *"Feelable but not seeable"* – Constraints and unknown context. The context of a haptic experience can limit the designer. Eyes-free interaction in cars means not only that visual reinforcement is unavailable, visible movement must be completely avoided. P4 is tasked with creating a *"feelable but not seeable"* sensation to *"avoid having to use visual feedback"*, as *"driver distraction is always a* *big topic"* (P4). This means P4 does not have full control over his designed haptic sensation, but can use audio reinforcement or substitution to handle constraints.

Worse is when the experience's context is unknown. Stakeholders often keep key contextual information secret from designers, such as the visual interface: *"we can suggest components, and suggest characteristics of the HMI [human-machine interface] system, but the exact visual design of the HMI system is the OEM's [original equipment manufacturer's] knowledge"* (P4). P3 has an *evaluation kit* to send to potential customers when customers' intellectual property (IP) is a concern:

> *"[An evaluation kit is] basically a little box that consists of our actuator and some electronics, and that box is connected and driven through the USB port of a computer, and you can then mechanically integrate the box in your own way, so we don't need to know what their design looks like"* (P3).

Unknown context can derive from at least two sources: IP and secrecy, and unconstrained end-user situation. We discuss IP and secrecy more in Section 8.2.4/CC3. Meanwhile, designers must deal with sometimes unknowable end-user context, especially with mobile scenarios. A high quality LRA-type actuator on a metal table can sound cheap, while an affordable eccentric mass actuator can sound like purring if it's on rubber, and *"there's not much you can do from a haptic perspective, other than allow the user to turn it up or down"* (P5).

Ex5: *"Very individual"* – Tailoring and customization. Because the context of haptic technology can vary to much, haptic designs need to be tailored for each client's problem and are often made customizable for individual users. For the former, several participants' business models are directly based on tailoring. P4's group makes a small set of actuators, adapting them to each specific request. This is exacerbated because it is *"hard for [customers] to really express what they need"* (P4) (discussed more in Section 8.2.4/CC1) so designers must rapidly and collaboratively fine-tune their solutions.

Even if customer goals are clear, tailoring is necessary because of requirements (e.g., branding or *"trademark"* (P2), see Section 8.2.4/CC4) and hardware setup: *"it's important to tune the experience depending on whatever kind of motor they decide to put in"* (P5).

> *"Depending on the outer design, what's given to us by the customer, we have to choose the direction of movement. For some applications, for some ideas, it's possible to move the surface directly perpendicular, away from the user, and other applications, you have to move the surface perpendicular towards the user, so the same actuation module could feel completely different"* (P4).

Meanwhile, individual differences further complicate matters: *"feeling right is...something that is very individual"* (P2). As P5 mentioned, volume controls can help end-users and adapt to unknowable context.

8.2.3 Collaboration: "Rally the ecosystem"

All six participants indicated collaboration was an important part of their work and design process. Haptic designers are part of interdisciplinary, international teams, and do not make haptic experiences alone:

> *"We basically have to rally the ecosystem...we have to go and find, y'know, somebody to supply the amplifier part, somebody to make the motor, somebody who knows enough about the Android kernel...we have to be, kind of, renaissance men if you like"* (P5).

In this section, we describe the collaborative ecosystem. First, we provide an overview of group structure and interdisciplinary roles found in our participants' groups (Co1), including a focus on the role of engineering (Co2). We then discuss the dispersion of stakeholders internationally and in different organizations (Co3), including a focus on a connecting role, sales reps (Co4), and the use of demos and documentation (Co5). We distinguish Collaboration themes (Section 8.2.3) from Cultural Context themes (Section 8.2.4) by focusing on specific communication methods and roles rather than underlying values and public consciousness.

Co1: *"I'm not so much of a psychologist"* – Internal roles are interdisciplinary. Haptic design is interdisciplinary; hardware, software, psychology, and business all play a role. P5 describes his company's job as *"rallying the ecosystem"*, finding diverse expertise and establishing a production chain. P6 describes different roles in her team, who work more closely together at different stages: *user [research], design, ergonomics, haptics, electronics, all come together* (P6). This is reflected by the diverse internal roles (Table 8.3), but also in the diverse work in single projects and individuals:

> *"We do some mechanical integration work, we help [our international customers] with designing the electronics, we have reference designs there, we have a couple of reference effects, and then we ship the part back and they go on with further doing the software integration and designing the haptic effects"* (P3).

Our participants worked in groups of various sizes. P2 worked with a student in a team of 2, while P5 describes different: design team, UX team, engineering, each

Table 8.3: Internal roles, the various descriptors used to label them, and descriptions.

Role	Descriptors	Description
UX	User division (P6), User research (P5), Ergonomics (P6), Human factors (P1), Psychologist (P2,6)	Related to design but separate role: *facilitate prototypes, validate, communicate those results* (P5). Here we've included psychologists and human factors roles because they conduct user research such as evaluation: *psychologists there who do usability tests* (P6), *study how effectively how users interact w/ goals* (P1).
Design	Design team (P5)	Related to UX but separate role. The design group ideates and communicate vision, developing a value proposition. Designers usually have a similar background to the UX group.
Engineering	Technical (manager) (P3), Engineering (P3,5) Electronics, mechanics, technical team (P6)	Often a separate division, handling prototyping and implementation (P5). They might test components, do physical construction, take requirements from design, ergonomics, electronics, mechanics, etc. and generate required (haptic) feedback (P6). This can involve both hardware and software.

with different responsibilities. This collaboration can be collocated or remote: P6 describes the different divisions in her company as being physically close together, while P3 has sales reps overseas to help with external collaboration.

Especially in smaller groups, team members fill multiple roles. Sometimes this falls naturally into their background: *"I guess [phone vibrations are] similar to mechanical control design, except that it's all virtual"* (P5). Otherwise, this lack of expertise leads reduces confidence: *"I don't know, I'm not so much of a psychologist to really, to dare to say I can evaluate subjective responses to tactile feedback"* (P2).

Co2: *"Go through the technical levels"* – Engineering support. Larger groups were able to have more specialized individuals. Especially common was a dedicated engineering or technical support team, tasked with implementing prototypes for design and user research teams.

> *"In our design research team we don't do any internal prototyping, we rely on engineering resources to do all our prototyping"* (P5).

P5's group says that neither the design team nor the UX team build prototypes, though the UX team facillitates and evaluates them. P1's team is similar: *give qualitative feedback and ranges to the technicians* (P1). Engineering departments are sometimes *physically very close to other depts* (P6), presumably to interact with different divisions and groups. However, separating expertise can cause gulfs of collaboration:

> *"If you try to go through the technical levels from a technology scout to a technical manager and then maybe to a senior manager, you usually get blocked with something, because nobody wants to take the risk or the blame"* (P3).

Those in engineering roles are risk-adverse: *"[it's] risky to suggest changes to their component"* (P3). P3 says that to pitch to other companies, you need to reach *"C-level people"* like the CEO, or other business or manager types: *"engineers look at it from a perspective well I'm going to take a risk if I change something in my design, and if it doesn't work everybody's going to blame me"* (P3), *technicians won't give pushback if there is a problem* (P1).

Co3: *"Different divisions, different companies"* – External roles are international. Haptic designers also worked closely with external stakeholders like potential customers and manufacturers. Our participants have suppliers, especially hardware suppliers, and often sold to manufacturers who then sell their own product to the end-user. Table 8.4 provides details on the external-facing roles.

> *"Automotive is very much a tiered and compartmentalized manufacturing business, and so the person who makes the control surface is different than the person who makes the mounting for it...and those people often never talk to each other, and so for us it's even worse than different divisions in a company, it's different companies"* (P5).

Often these groups are distributed internationally. P5's group, based in North America, received international demographics to research: *"here's phone X from OEM Y and it's targeted at Asian ladies from 15 to 30 years old"* (P5). P3, who has a headquarters in the North America but has clients in Asia, describes *sales reps* as critical team members who can bridge language and cultural barriers.

Co4: *"Sales reps"* – Facilitators and advocates. P3 describes sales reps in-depth as key team members. Sales reps are trained locally at the headquarters in North America, then sent the customer's area, often countries like Korea, Japan, China, and Taiwan which have large consumer electronics and gaming markets. They can speak the local language and understand the local culture, but also facilitate demos and persuade customers to pursue business with the designer's team. If a demo is sent to a company without a sales rep, customers sometimes respond by shipping the device back and requesting assistance, but often don't respond at all:

Table 8.4: External roles, the various descriptors used to label them, and descriptions.

Role	Descriptors	Description
Connections	Sales rep, technology scout (P3)	Sales reps from haptic companies handle local expertise (language and culture), haptics expertise (they run demos), and can be advocates for products. Technology scouts from large companies talk to haptics companies to learn their technology.
Business	Business development people, C-level people (P3)	Internal business development people are *"here [in HQ]"* (P3), while external business people make decisions; they're who you need to persuade, rather than technology-focused roles.
Supply chain	Vendor, developer, manufacturer, OEM (original equipment manufacturer) (P5), supplier (P4,6), content provider (P3)	Haptic designers are heavily embedded in a supply chain involving hardware and software manufacturers. Some manufacturers provide hardware (e.g, actuators) and software (e.g., Android API) to the haptician, others are the intended customer (phone or car manufacturers, software developers). It is unclear who creates haptic content in this ecosystem.

> *"If we try to just ship them a part...in the best case they come back and say well it doesn't as we thought, can you help us?...in the worst case they don't even contact us back and we never learn why they didn't pursue an idea or an opportunity. It's still a complicated setup to make haptics work, there?s lots of aspects that you have to take into account, and if you don?t do it properly, you're going to be most likely very disappointed about what the outcome is"* (P3).

Big tech companies sometimes invert this from a push model (where the haptics company uses a sales rep) to a pull model with tech scouts (who reach out to haptics companies). Sometimes, companies fill this role without dedicated sales reps: P4 goes to customers regularly in confidential meetings, receiving specifications and working collocated with the customer to get their product to feel *"just right"*:

> *"There is always [the] option, as we did with one of our customers, that we simply went into the lab for a day or two, and just worked on simulated button feel, together with the customer, to get the feel just right"* (P4)

In all cases, content can fall through the cracks. P3's company provides technology, but *"the issue that we are having with uh, the content providers that need to* *get interested and believe into it...creating the haptic effects is something that we haven't been involved in in a lot of detail in the past"* (P3). P5's company does have a set of 150 effects, from which they select themes. The other participants all mention technology they develop, with content directly related to their hardware solution.

Co5: *"Your piezo demo, we love it"* – Demos and documentation. Demos are essential to showing both the value of a haptic experience and enabling two-way communication with the customer. They can clarify requirements and grab attention from clients: *"we'll often get the OEMs who will say, well you showed us your piezo demo, and we love it, it feels great"* (P5). Demos can be conducted in-person (synchronously) at events like tech-days or one-on-one meetings: *"the customer either comes directly to us, we go towards our customers regularly, have our tech days, similar to automotive clinics"* (P4), or asynchronously, remotely shipped. Demos are complicated and need an experienced handler like a sales rep. Once setup, demos are often adjusted, but this is easier than setup: *"From the moment the actuation module was working...it was just cranking up the maximum current or reducing the maximum current"* (P4).

Demos are often collected into groups. P5 describes downloading apps that use his technology and *"sticking those in [their] demo suite"*. P1 and P2 talk about collecting examples for inspiration and guidance early in design: it's *quicker to go out and buy examples*, like *15 or 16 appliances that had notably different feelings* (P1). P2 instructed his student to *"collect physical push buttons just to get in contact with all the diversity of stuff"*, and ended up with a *"button board"* to guide design. He also talks about company guidelines:

> *"When I was at [a major automotive company] 3 years ago...they had this guideline book...they had guidelines on the design of physical widgets like sliders, physical sliders, push buttons, rotary things...they defined thresholds basically where these forces have certain thresholds and if you get over the threshold something is happening"* (P2).

Demo setups can thus be stored long term for internal documentation (button board, guideline book), but they can also be ephemeral (tech days). In both cases, they can help to articulate the value, especially valuable when most people do not yet understand haptic technology.

8.2.4 Cultural Context: "A standard feature, in the future"

Haptic technology has yet to penetrate the public consciousness. Participants reported major difficulty when working with both customers and users, including a lack of understanding what haptic technology is and how to work with it:

> *"People really don't know what to do with [haptics] and I think within the haptics community we need to...continue to push it into the market, but once it's there I think it's going to add to the user experience and will be a standard feature in the future"* (P3).

Specifically mentioned were the difficulty in understanding customer requirements (CC1), and knowing how to appropriately evaluate haptic experiences (CC2). As a technological field, secrecy and intellectual property are important concerns for both designers and customers (CC3). Designers had ways to pitch the value proposition of haptics, often tied to UX and branding (CC4), but risk and cost of adopting the technology often make it a hard sell (CC5).

CC1: *"Hard to express what they need"* – Understanding requirements. Customers found it difficult to both understand and request their needs. Our experts focused on the end result because it gives them and their designers the ability to solve problems: *Don't specify elements. Only give end product. Don't tell how to restrict; can give hints* (P6). However, requested end-results are often vague or confusing, like *"good variable feel"* (P4):

> *"The customer only came with a question, yeah, how does it feel variable? (laughing) Here it did not really describe how it should feel variable"* (P4).

To make these impressions concrete, customers initially give engineering parameters as their best guess. P4 in particular talks about his customers, who might point to a *"reference button which is available directly on the market, from companies like (company 1) or (company 2), and they say it has to feel exactly like this button"*, or request *"a surface acceleration of 10 to 20 G perpendicular and a travelling distance of .2-.3 mm"* (P4). This might have little bearing to the final result, after the designers iterate with the customer: *"we ended with an acceleration of 2G and a travelling distance of .4 of a mm,? so, due to the size of the module, simply the high accelerations were too high for a good variable feel"* (P4). The goal function of good variable feel was achieved, but the initial engineering-level specification was completely off.

Other participants showed this duality between high-level affective goals and low-level guesses. P1 especially used affective and psychological terms when considering design, such as semantic differential scales: *good/bad; gender (robust/delicate; size); intensity (sharp/dull; bright/dim, fast/slow); novelty* (P1). Haptic designers often connected low-level/high-level terms through iteration, or with their own way of representing features like quality: *"[audio click gives] quality, and, consistency across the whole dashboard"* (P5), *mass is big for quality...for the haptics, nice feedback w/ good snap gives a sense of quality* (P6).

CC2: *"It feels right"* – Evaluation. Our designers all evaluated their designs but demonstrated different methods of evaluation, consistent with our workshop survey (Section 8.3). P2 explicitly evaluates both low-level, pragmatic concerns (e.g., , task accuracy and speed) and high-level affective concerns like feeling ?personally involved? (with the AttrakDiff questionnaire [**?**] (http://attrakdiff.de)). P5's user experience team conducts validation (details aren't included). Small-scale acceptance testing was employed by both P2 and P4: when iterating in-person with the customer, P4 kept iterating until the customer said it *"felt right"*; P2 only had himself and his student evaluate their designs in an academic context, despite indicating a desire to do a more thorough evaluation. P3's group doesn't create content, but indicated a desire to look into that and investigate it with studies.

There seems to be a clear desire for stronger evaluation, but as-is, it's mostly lightweight, ad-hoc acceptance testing. This is consistent with our workshop findings, which suggest little real-world or *in situ* evaluation. However, deployment seems to be a natural way to see if the design is good enough, as the ultimate acceptance test. P5 described his most memorable moment of his software project being when his product had been deployed and used by a software development team. Seeing a haptic-enabled app available for download, and feeling it in context, was impressive:

> *"I think the most memorable day was when we started downloading apps, and realized that, yes, in fact this does work, and not only does it work but it works pretty well for a variety of apps... we ended up just sticking those in our demo suite even though we had no relationship whatsoever to the developer. So, their app just worked, and it worked really well"* (P5)

CC3: *"Kept confidential"* – Secrecy and intellectual property. Sometimes the customer does not know what they want, but in other cases, they have important information they need to withhold. As mentioned in Section 8.2.2/Ex4, secrecy in haptics has major implications that inhibit design, especially given the verticality of haptic technology:

> *"Somebody wants to design a completely new gaming controller for a gaming console, so they might just have some CAD drawings or they might have something they don't want to share with us, so in that case*

> *we provide them an evaluation kit...we don't need to know what their design looks like, they can really work on it internally"* (P3)

P3's clients are able to receive an *evaluation kit* and create content with audio editors. P4 describes meetings with customers that preserve confidentiality: *"on these tech days it?s usually only one customer and not that many suppliers at the same time, sometimes only the customer and us, to make sure our development are kept confidential"* (P4). Once technology of P4's company is on the market, it's no longer secret - rivals can copy or reverse engineer the devices, so there are lots of demonstrations before release of the tech.

Secrecy can cause delays for software too. P5 delivers a modified Android kernal to his customers, software developers. However, they aren't given an early release, and thus they *"always lag the market by two months at least, to get an update [for Android]...it?s annoying because, y?know, as soon as the OEMs get the source code they want to put it in their product right away"* (P5).

CC4: *"Articulating the value"* – UX and branding. Our participants were all passionate about haptic technology and its benefits. The value of haptics can be connected to better performance on various tasks: P2 tried to *"support people interacting bimanually to find out if they are more accurate in drag and drop tasks, [or] faster"*, but also whether they would *"feel more personally involved in the interaction somehow"* (P2). This latter goal, of user experience or rich feedback, was the primary value for haptics:

> *"It's? like having a touchscreen now on smartphones which nobody expects any other way anymore...sometimes pull out my old, uh, tom-tom navigation device in my car, and that one didn't have a touch screen back then (laughing) so I tap on that one [expecting it to respond to touch input], and so it's the same thing with haptics, at some point it?s just going to expect that you get some nice haptic feedback, but getting there is still a couple of years out"* (P3)

As mentioned in Table 8.2.2, tailoring and customization are important for each implementation. This is also true for value: differentiable sensations are important to help distinguish overall user experience and provide branding. *look for alarms that were different; branding. effective, but different.* (P1). Companies and products need to have both a cohesive and differentiable feel. P2's company *"guideline book"*, which defined force profiles for buttons, was helpful to *"coin a trademark"* (P2).

> *"[We] provide differentiated tactile experiences to our customers, who are major mobile phone manufacturers. Since Android is pretty generic*

> *across the board, um, they like to have custom themes, which are sets of these 150 effects"* (P5).

With software libraries, themes are essential to the haptic design process. This desire for consistent output has a tension with customization and fine-tuning: *"it's also important to tune the experience depending on whatever kind of motor they decide to put in"* (P5). This is part of the persuasive capability of haptics: *improve comfort and differentiate based on branding* (P6).

CC5: *"A tough sell"* – Overcoming risk and cost. Despite its value, haptic technology is a risky, costly feature to add. Providing improved user experience requires *"high-definition haptics"*, not *"some rumble feedback that has been around a long time"* (P3). This often meants *"going up fidelity"* from a *"cheap, poor quality motor"* (P5). P5's company argues that *"the end-user is going to prefer this quality of experience"* with improved hardware, like a piezo actuator.

> *"[If we were to perform this project again,] I think we would spend a bit more time up front articulating the value, the specific value prop, of individual features"* (P5).

P5 notes the challenge of convincing non-end-users to buy or deploy their technology: *"[our company] has the unique challenge that our customers are not the people who use our products"* (P5). Since the main benefit is to UX, it's challenging to connect to the bottom line, especially compared to other haptics components. Designers need to *"get up to the decision making level and more on the business side...[business roles] know nothing about technology, I mean, they don't care, but we are trying to demo parts to them, present business cases to them, and show them what they can do in order to gain market share,? or increase their retail price when they add our technology"* (P3).

> *"Let's assume we were to work on a completely different product like memory chips, so? everybody understands what this is for, what it can do, and you probably have a memory chip that is faster or, whatever, smaller. Now for haptics, this approach is kind of difficult because the technology scouts themselves they kind of understand what this is for, but how it it's going to add value to their device, and how much they can increase the retail price, or if they can increase it at all, or gain market share, that's completely open"* (P3).

Newer technologies are hard explain: *"[gesture-based haptic feedback is] a much more complex task to design, and also to explain, to the OEM"* (P5). It can also make persuading a customer difficult. P3 finds that there is always *"there's always discussions on the cost"*, and despite *"proposing also alternative business models"* like contracts *"you pay over two years for your phone"*, *"nobody really signed up"* (P3). This is perfectly captured by P5:

> *"'We love [the piezo demo], it feels great, we're building this phone that has a 10 cent eccentric mass motor in it, can you make it feel the same?' The answer of course is no"* (P5).

P5 notes that *"cost pressures are pretty extreme"*, mobile phones in the US cost *"$199 on contract, that's sort of a fixed price? and you can add more features to the phone, but that just reduces the profit margin, right?"*, so *"the addition of haptic feedback technology...can be a tough sell"* (P5). Haptic technology is especially risky because of previously discussed challenges: it involves separate risk-adverse engineering divisions, and changes to the *"guts"* of a product. Designers need to set up complicated demos to persuade decision makers of the value of improved user experience: *if only compete on cost; then this is tough* (P1). Of course, *"it's hard to get through to the right level"*, like *"C-level kind of persons, so, talking to the CTO of Sony, those kinds of people"* (P3). The combination of high-risk, increased cost, and tenuous connection to the bottom line make haptics a very tough sell indeed.

8.3 Broad-Spectrum Follow-up: Workshop Findings

Here we report the findings from a workshop on haptic experience design at World Haptics 2015, a major academic haptics conference (http://haptics2015.org). The workshop was organized by some of the authors to initiate a conversation between researchers and industry practitioners about HaXD, and elicit feedback on our findings to date. Four leading haptic design experts gave talks which set the stage for a brainstorming session, culminating on a panel discussion about challenges and ideas for HaXD. Over thirty participated in a workshop brainstorm session and panel discussion. Sixteen participants responded to a questionnaire at the end, which requested details about respondents' roles, tools, and techniques (Section **??**). In the following, we supplement the quantitative questionnaire results with the qualitative data gleaned from this venue – open-ended questionnaire responses and themes from the brainstorming session and panel discussion.

8.3.1 Quantitative Results: Tools, Background, Groupwork

Of 16 respondents, 5 self-reported as working in industry, the other 11 as members of academia (one of whom also reported also working at "other: research insti-

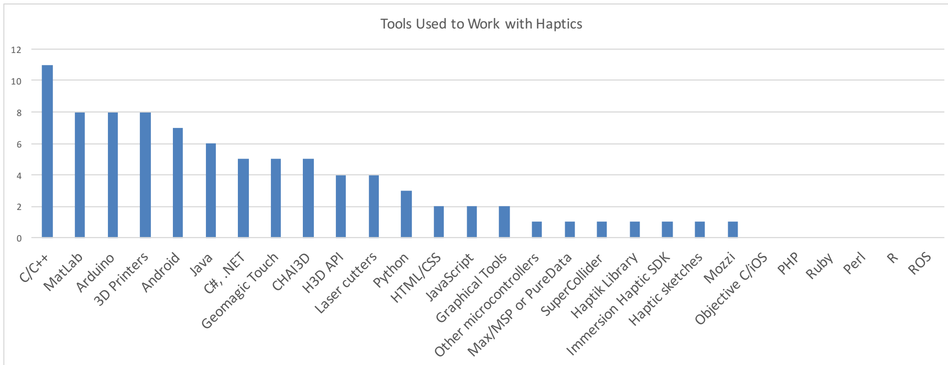


Figure 8.2: Responses for tools used in haptic design (N=16, *"check all that apply"*).

tute"). As roles, 4 reported as graduate students, 4 as developers, 2 as designers, 2 as a combination of designer and developer, 2 as researchers, 1 as a business person ("product integration/commercialisation"), and 1 did not respond.

Respondents reported a wide variety of hardware and software tools used to work with haptics (Figure 8.2). Most used were popular general or technical programming languages like C/C++, Matlab, Java, and hardware hacking tools like Arduino and 3D printers. Force-feedback APIs for consumer hardware (Geomagic Touch CHAI3D, H3D) were moderately used. Very few respondents reported using scripting or web tools, like Python, HTML/CSS, JavaScript, or more specialized tools. This combination suggests needs for for performance, technical or scientific libraries, and an ability to tightly access prototyping hardware; while (in contrast to may other media design domains) web tools use is notably low. The latter is not particularly surprising for design that is, by itself, not primarily visual, and often comes with tight timing requirements.

Evaluation techniques were also varied (Figure 8.3); many respondents listed several. Most common were methods deployed in-lab or in-house (piloting, laboratory studies). Less common but still used were more externally valid evaluations (*in situ* studies and real-world deployment). Quantitative and qualitative methods were reported with equal frequency: 8 participants reported using both, i.e., a mixed-methods approach.

Group size reports suggested that hapticians work in groups with varying sizes (Figure 8.4). Few work in large groups; just one person (the designer/developer for a research institution) reported a group size of 21-50, no one reported a group of size 11-20, and most reported working with 3-5 others. Five participants reported varying group sizes (combinations of 1, 2, and 3-5 people). Because our question did not precisely define the meaning of a "group", we note the possibility of interpreting it with differing degrees of collaborative closeness, but assume that it was generally regarded as meaning the sharing community within one's organization.

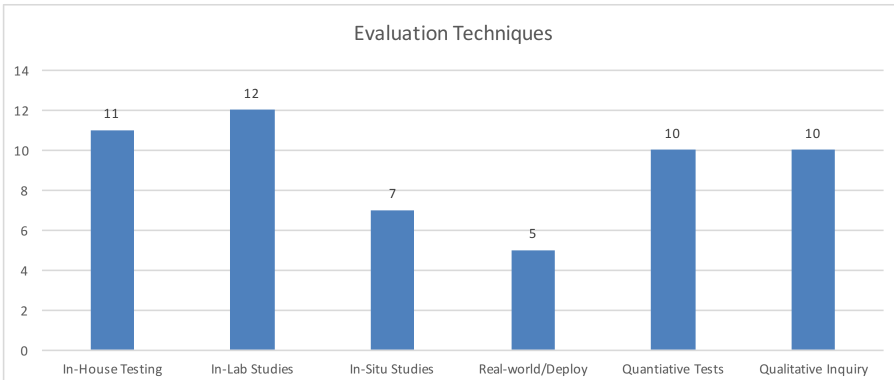


Figure 8.3: Responses for evaluation techniques used (N=16, *"check all that apply"*).

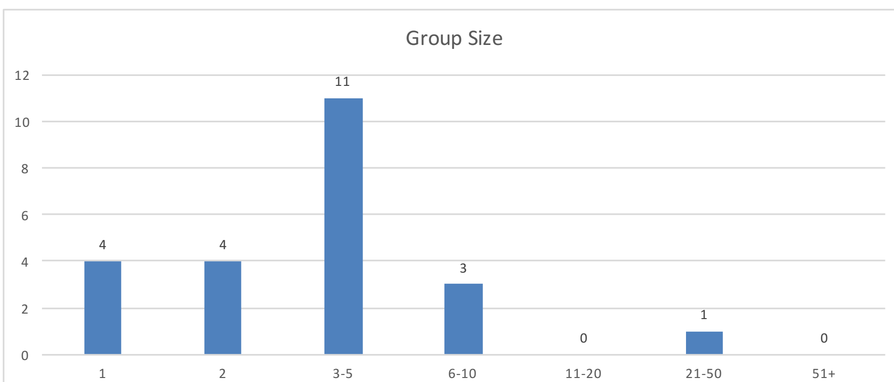


Figure 8.4: Reported group size for projects (N=16, *"check all that apply"*).

8.3.2 Qualitative Results: Consistency, Quality, Value

Qualitative responses from free response questionnaire questions highlighted three challenges for haptic design:

1. **Universal design**: *"universal experience"*, *"adopting wide spectrum of users"*, *"optimal and consistent delivery of haptic cues to large number of people"*,

"users variations in terms of subjective analysis", *"common haptic experiences, any person/any device"*, *"the spectrum of perception"*.

2. **Evaluating quality**: *"what is 'good enough'?"*, *"modeling haptic quality of experience"*, *"appropriate fidelity force feedback"*, *"optimal and consistent delivery of haptic cues..."*.

3. **Value**: *"getting people to realise the benefits of good haptics. Finding new ways to use hi-fidelity (wide bandwidth) haptic feedback to enhance UX"*, *"Bringing haptics to mainstream/consumer electronics"*, *"merging the technologies, make safety and pleasure experiences"*, *"convincing it's useful"*.

Other comments include emotion (*"transfer emotion through haptics"*) and language (*"haptic language; no simplicity in generating new sensations"*). When asked to what they would like to see in a design tool to overcome these challenges, respondents suggested ways of handling variability or ineffability, such as automatic configuration:

> *"mapping"*, *"automatic evaluation of systems and actuators"*, *"quatification [sic] of haptic perception"*, *"autoconfiguration, calibration and prediction of results with the users"*, *"accessible to all, supported by standards"*, *"autoconfigure depends that perception [sic]"*, *"bigger testing pool"*.

Brainstorming occurred in 6 groups, who tried to identify challenges faced by members of each group, then brainstorm solutions. Many of these challenges mirrored questionnaire results. Three groups (G1,2,6) focused on the *value* of haptics: what is the point or goal in certain situations, and how to market haptics. One group (G3) tackled the problem of *examples of good design*, including hardware and software architecture, suggesting a repository like GitHub or software engineering patterns. Three groups (G4-6) talked about *meaning* – and subjectivity therein, including the possibility of a shared or useful language.

The follow-up discussion developed these challenges further. Emergent themes were that haptics is not well marketed and that touch is taken for granted. The word "haptic" might be too jargonny or poorly understood; perhaps other terms like "tactile effect" or "physical effect" could be more useful. Curated examples and an online repository were offered as valuable goals.

8.4 Discussion

To synthesize takeaways out of our findings, we begin by describing critical activities practiced by haptic designers, showing that HaXD is already a subfield of design. We then identify major challenges encountered in HaXD that are unique to or exaggerated with haptic experiences. We conclude with several concrete recommendations to support HaXD in the future, and a vision for what it might look like.

8.4.1 Activities of Haptic Design

From this work, and previous work on HaXD tools, we suggest the following activities that haptic designers conduct, all familiar to designers in other fields.

Develop and communicate vision. Designers must articulate the value that their designs can bring to both end-users and customers. They must communicate value to their team and others, and critically, **persuade external stakeholders** that their product will contribute to the bottom line. To do this, they must **collect, run, and tune demos**, a critical part of the communicative toolkit for haptic designers.

Prepare for design. Designers need to **divine requirements** from customers, which customers often do not understand themselves. Designers also **gather examples**, both to provide inspiration and facilitate communication. Designers need ways to capture, modify, manage, find, use, and share examples and ideas, both ones they develop themselves and ones they seek out for inspiration.

Iteratively develop multiple concepts. Our designers needed to iterate, often with their clients' or users' feedback to really find the right designs. Design thinking and UCD are both important to apply to haptics, especially because requirements are difficult to understand. Additionally, designers must either **communicate with engineering**, articulating requirements to receive new physical prototypes, or **have engineering skills** to create demos and prototypes themselves. During iteration, designers must also **evaluate designs and collect feedback**, although this practice is currently limited. Both informal feedback from colleagues, and formal studies, typically run by a UX/research division.

Interface with research. Designers need to hand off prototypes or stimuli to their user research division, and communicate study goals. They must also interpret results from studies from multiple sources: marketing research, feedback on prototypes and stimuli, and monitor the academic research in this rapidly changing field. Alternatively, as with engineering, they might **plan, run, and analyze studies** directly.

Manage IP. Designers need to be sensitive about intellectual property, both that of their company's technology, and of the many companies and divisions they interact with.

8.4.2 Challenges for Haptic Design

From our participants' themes, we identify several challenges facing designers that are unique or exacerbated when working with haptics.

C1 Context is Unknowable. Haptic experiences are multimodal and vertical, interacting closely with physical hardware, grip, and orientation. When our participants knew the haptic experience's context, even when it gave constraints like avoiding visible feedback, they were able to use reinforcement and substitution to improve their sensations. However, when the context is unknown, either from not having access to relevant intellectual property or due to diverse environments and grips, designers are very limited, especially when trying to create consistent experiences. As P5 said, there's not much you can do other than assigning a volume control.

C2 Individuals Vary. Another unknown factor is how the user perceives, understands, and responds affectively to a haptic experience. Different people, who experience perceptual differences based on age, sex, and their need-for-touch, will have different responses to a haptic sensation. More over, their past experiences affect how they will interpret a sensation, e.g., feeling a vibration as a cat's purr or a car's rumble. This has both an effect on end-users and on customers, as each application area needs the haptic technology to be tailored, hindering production at scale.

C3 Demos are complex, costly, and crucial. Despite being essential in communicating with stakeholders and persuading customers, demos are hard to set up (many moving parts and ways to fail), and often require a dedicated assistant. Latency is a special challenge for early prototypes, which can defeat the prototype's objective of trying out the experience. However, haptic design takes place internationally and through different organizations; between secrecy concerns and remote collaboration, we can't always have someone present to nanny a demo. Once set up, however, they seem to be effective, especially if they are tunable or customizable.

C4 Iteration is painful. Every change to a haptic experience results in a change to the "guts", including reinforcing modalities and physical setup. Tight technical constraints, like low latency, need to be followed to be effective.

This means that iteration is slow and difficult. However, customers and users don't understand requirements without trying ideas, and haptic technology must be tailored to each application. Thus, rapid iteration or parallel development is essential to get things feeling "just right". Despite this, iteration remaines painful; special care is taken by our designers to allow for easy tuning, but better solutions remain to be seen.

C5 Cost/benefit ratio is not persuasive. Haptic technology, ironically, often provides intangible benefits: better user experience, branding, and sense of quality. Designers are tasked with persuading phone and car manufacturers of better user experience, rather than users themselves. Improving haptics comes with increased cost and risk, often is communicated through risk-adverse engineering avenues; all the while, it is hard to relate to the bottom line of increased revenue or profit. Lack of infrastructure, public knowledge, and ability for people to say what they need means we need to establish haptics more prominently in the public consciousness.

C6 Barriers to interdisciplinary collaboration are significant. Haptic design is extremely multidisciplinary. Because of this, our designers either need to fill many roles, or work in groups that include hardware, software, business, and psychology. Making skills, including engineering, are especially valuable but often segregated from the central design group. Especially because haptic technology needs to be tailored to each problem, we cannot design once and forget and need to maintain multidisciplinary groups.

C7 Evaluation methods are limited and often inaccessible. Haptics provides intangible benefits like better quality of experience or branding, which are difficult to study. Further, although many of our designers mentioned evaluation methods as important, time constraints limited how much evaluation is conducted. Typically, acceptance testing seemed to be the primary result. In our workshop, we found both qualitative and quantitative methods were used, but *in-situ* evaluations were difficult to come by, suggesting that haptic designers primarily conduct evaluations in-lab and do practical deployments

8.4.3 Recommendations for Haptic Design

From these challenges, we identified several directions for development and support which could lead to better haptic design in the future.

R1 Closed loop control for consistent performance. To guarantee specific perceivable outcomes in an unknown environment (C1), sensing and control theory can be applied to drive actuator output to desired levels. For example, by sensing the external vibrations of a VT actuator with a microphone, its output can be modified to overcome the effect of external factors like material, orientation, and grip and achieve a specified frequency, amplitude and responsiveness.

R2 Easy demonstration and sharing. When exploring, demonstrating, and sharing designs is easy, iteration and convergence is faster and best practices propagate – e.g., , evaluation kits where customers and clients can experience demos without a sales representative being physically present to orchestrate the experience. But demos that are self-assembled or operable by non-experts require an easy way to troubleshoot and ensure calibration. We therefore recommend development of automatic or facilitated calibration of demonstrations to support unknown contexts (C1), individual differences in perception and preferences (C2), and streamlined demo sending, setup and quality assurance (C3).

R3 Customizable infrastructure. ⌈**KM** *slc: conflict with rapid prototyping ethic?*⌋ Individual differences persist even when context is known (C2). We recommend adding customization tools whenever possible, to select options, fine-tune and simply adjust volume. Customizable infrastructures that support fine-tuning can also help speed iteration once demos or even fully-fledged applications are set up (C4), letting designers and customers try variations of a haptic experience easily.

R4 Proxies and pre-feels. Proxies are another way to address the complications of remote demos (C3) – e.g., , low-fidelity feedback like phone vibrations when high-fidelity feedback is unavailable [**?**]. Low-fidelity previsualization of haptic sensations (or "pre-feels") [**?**] can improve iteration speed (C4), by allowing the designer to experience an approximation of an iteration before committing resources to building it, and/or to compare with a reference starting point.

R5 Hardware pre-simulations. The speed or effectiveness of *mechanical* iteration (C4) can be improved by using software simulations to explore how different electronics components could be rearranged to preserve or enhance dynamics. Even more advanced might be the use of simulations to develop "perceptually transparent" sensations [**?**], allowing actuators or other components to be swapped in and out if upgrades or cheaper models are available, while software components are automatically updated to achieve a consistent end result. This might be best combined with closed-loop sensing

(R1) to ensure simulation accuracy.

R6 **Develop infrastructure.** As hinted at with other recommendations, we need to develop better infrastructure to explore and deploy haptic experiences. Especially to improve the painful iteration cycle (C4), we suggest building infrastructure for developing modular systems, customizable or adjustable demos or environments, and large online repositories of examples, from which we can adapt experiences and build upon previous work. Broadcasting haptics remains an important and unrealized goal, which can help both with potential customers (C3) and end-user experiences.

R7 **Outreach and education.** Another way to improve perceived value of haptics (C5), and to facilitate the interdisciplinary communciation required by haptic design teams (C6) is to perform more outreach. Having public portfolios, accessible haptics education, support for DIY and maker cultures, and events like hackathons will help to establish haptics as a known term, spread the word about its value, but more importantly help more people join the conversation to find the value in touch-based technology. It will help provide different stake holders with common reference points, language, and understanding, both lowering the bar to conduct haptic design as a team member, and by providing a voice to external stakeholders.

R8 **Design theory language.** Language especially will be important to develop to help the different team members and clients communicate (C6). Much like graphic design, where non-experts might be aware of some concepts (symmetry, contrast, hot/cool colours), and experts know much more (colour combinations, concept of weighting in a visual design) will help teams communicate across divisions and with clients, users, and customers. It remains to be seen whether this will be a formal lexicon of terms, or ideas that emerge organically; either way, we suggest paying careful attention to the language used when doing haptic design, and to share the language alongside the sensations and their components.

R9 **Better evaluation tools.** Haptic designers have limited access to evaluation tools and techniques that are taken for granted in other modalities, especially *in situ* tools (C7). Promising ways of mitigating this handicap is application of remote analytics to haptic design, e.g., logging, machine learning, or qualitative contextual inquiry. This may require development of new batteries of haptics-suitable tests, especially ones which target its less objective benefits (e.g., , quality and branding), which might in turn help to study perceived value and risk (C5).

8.4.4 Future of Haptic Design

Our interviewees followed an observable, defined process. They collected requirements, developed multiple concepts, and iterated until they arrive at a final experience, which is then evaluated with varying amounts of rigour. We saw evidence of libraries, examples, and our participants' own craft and experience; we also see a diverse, international, collaborative ecosystem. Some deliberately applied user-centered design.

However, we also see that haptics *"might be 30 years behind graphics"* (P3), or at least *"really new"*, i.e., in an early stage of development. A vision of HaXD as a fully-developed design subfield is exciting, but challenging. Experience design in general is still finding itself. However, we believe that HaXD can draw from both newer fields like experience design and interaction design, as well as more established fields like graphic design. How might it look?

First, haptic designers might work in teams, interacting with other relevant units: creative, product design. In fact, from our research, it is likely that haptic designers will need to be vertical, communicating with everyone from mechanical engineers and software development to people conducting business and user research. As with graphic design schools, there may be formal education available for haptic designers. However, as haptic technology needs to be tailored to each specific problem, these are more likely to be generalized professional programs that train diverse skills, or will focus on certain sub-categories of haptic technologies, e.g., , tactile artists or animators [**?**], friction designers, or 6DOF force-feedback developers. As hardware becomes more affordable, we also expect the recent makers movement will encourage hobbyists and artists to explore haptic technology and push its limits, possibly in the near future.

Like other emerging media, e.g., , the web browser wars of the 90s, standardization of HTML/CSS, and Blu-Ray vs HD DVD, we expect various file formats and infrastructures to emerge and then coalesce. Given the diversity of haptic technologies and experiences, we expect these to be centered around *paradigms*. For example, haptic icons [67] are one paradigm: display-only, temporal entities rendered on a single location; these might be designed, distributed, and experienced similar to audio files. However, tactile animations [**?**] are another paradigm: generalized spatio-temporal entities that can be rendered continuously on different grids. Multi-DOF force-feedback displays are often programmed with a third paradigm: a virtual environment and a single manipulator; this is most analogous to 3D virtual worlds. Paradigms can be applied to multiple devices in a class (e.g., tactile animations on grid displays), or multiple paradigms might apply to a single device (a Haptuator [110]) can display a haptic icon (temporal only), or it can produce a directional force [**?**] (spatio-temporal).

We expect design dimensions to be further developed, and eventually encapsulated into best practices like alignment, contrast, and weighting used for graphic design. Other design languages, like musical notation, will facillitate recording and communication amongst experts. Meanwhile, more developed aesthetic theories, like musical theory or colour theory, will help guide people to effective, pleasing, differentiable haptic designs. Intellectual property law will need to be adapted - much like a logo can be trademarked, how might a certain button click? Whether an haptic icon set be protected, and how to set an appropriate level for burden of proof, remain open questions. We hope that these questions and more will be answered during this exciting time for one of our most essential senses.

Chapter 9

Conclusion

In this dissertation, we explored haptic experience design (HaXD), looking at its process to inform how to design, build, and evaluate HaXD support tools. We leave the specific findings from each of our studies as best articulated in their respective chapters. In this chapter, we summarize larger trends across these chapters and give some final thoughts about the HaXD process and support tools. We begin by discussing process, including challenges and strategies designers can use to overcome those challenges. We then discuss findings more specifically on designing and developing software tools to support HaXD. We conclude with directions for current and future work, and final remarks on HaXD.

9.1 Summary of Research Findings

Through Chapters 3 to 5 we designed, built, and studied a trio of HaXD support tools. We began with an initial hypothesis: that real-time feedback and collaboration could improve the haptic design process. Through our tools, this hypothesis was confirmed, but more importantly, elaborated and refined:

- mHIVE showed us that rapid exploration was possible with real-time feedback, demonstrated value in informal, collocated collaboration, and gave evidence that showing rather telling about haptic sensations could circumvent an impoverished language (e.g., what do you think of *that*).
- mHIVE also showed us that there are distinct roles to be played by different tools: mHIVE was successful for early sketches, but did not enable refinement and had a learning curve.
- Mango followed up on these themes with a focus on implementation. We established a rendering pipeline for both design and playback. A direct-

manipulation metaphor let participants both sketch in real-time, and refine their designs. In addition, an animation paradigm enabled our visual animators to transfer their skills to tactile animation.

- In our Mango studies, we found evidence for reuse - repetition played a large role in tactile animations, and participants again drew inspiration from their experience or external examples (e.g., a YouTube video of a heartbeat) - and noted the engagement of multimodal design, e.g., designing for an audio clip. These both informed Macaron.

- In Macaron, we were able to easily implement the system (drawing from Mango's architecture), allowing us to study our process more closely. We found a number of concrete recommendations for HaXD tools, and confirmed the value of using examples: we found different strategies for using examples in the initial design process, and open or visualized examples helped designers learn how to conduct VT design.

- Macaron also helped us find our more nuanced understanding of our initial hypotheses. Real-time feedback is useful as a preview, to get the right frequency, amplitude, or timing. However, participants would also step back to feel the entire design in its entirety, something that can be done

Together, mHIVE, Mango, and Macaron have informed us on both important features and roles for design tools, given us insight into implementation and evaluation, and helped to study HaXD as a process. To further our understanding, we conducted several focused projects which broadened our understanding of collaboration (Chapter 6) and different devices and application areas (Chapter 7):

- HapTurk (Chapter 6) allowed us to study collaboration more thoroughly, as we focused on other aspects of design with Mango and Macaron. We found that VT icons could be deployed over MTurk to elicit large-scale feedback from remote users. We also found affective qualities of VT icons could be communicated through proxies, suggesting we can express haptic ideas in different modalities.

- FeelCraft and Feel Messenger ...

Finally ⌈**OS** *OS resume here*⌋

9.2 HaXD Process: Requirements for Tools

In this section, we examine our findings about the HaXD process. In Chapter 8, we saw that haptic designers follow a familiar design process. However, we also saw that there are unique challenges that differentiate HaXD from other modalities of design, which are confirmed by our work into HaXD support tools and our design side projects ⌈**OS** *need to standardize on the name*⌋.

We begin by discussing four design activities that occur generally in design, but need to be explicitly supported for HaXD: browse, sketch, refine, share. Then, we comment on some approaches for handling the diverse devices and modalities employed by haptic technology. After, we discuss some techniques to imbue haptic experiences with meaning and realism. Finally, we talk about the importance of customization and how to support it.

9.2.1 Contextual Activities of Design: Browse, Sketch, Refine, Share

In our first exploration of HaXD support tools, the Haptic Instrument (Chapter 3), we found evidence that mHIVE was able to support exploration and sketches of haptic ideas, but not refinement into final designs. mHIVE was also able to support collaboration in certain ways. In our followup studies, we explored these activities that draw upon a designer's context, eventually arriving at four that we found are valuable when thinking about tool design: browsing examples, sketching new ideas, refining existing ideas, and sharing ideas with others. These activities can occur at any point in the design process, and we do not propose them as an exhaustive list; for example, "framing" [93, 106] could be claimed as an activity for design which may overlap with activities like Browsing and Sketching. We focus on their utility in motivating features and specific tools to aid haptic experience designers. Figure 9.1 shows how these activities fit in the design process.

Browse

"Browse" can have specific meanings for interacting with data (cite Munzner). Here, we use "Browse" to refer more generally to the act of looking at examples (e.g., , corkboards of previous designs [10], drawing from previous personal and professional experiences (e.g., one's repertoire [93] or real-life sources to inform a design. We found this emerged in different ways: in Chapter 3, participants used their personal experiences to interpret sensation meaning (i.e., schemas, discussed more later); in Chapter 4, one animator brought up a YouTube video of a heartbeat to ground his animation; in Chapter 8 several participants described collecting examples or using guide books.

We highlight this activity because haptic designers encounter modality-specific barriers when gathering, managing, and searching for examples. Explicitly supporting browsing can make a difference: in Chapter 5, we found visible, incorporable examples both eased design and helped scaffold learning for non-experts.

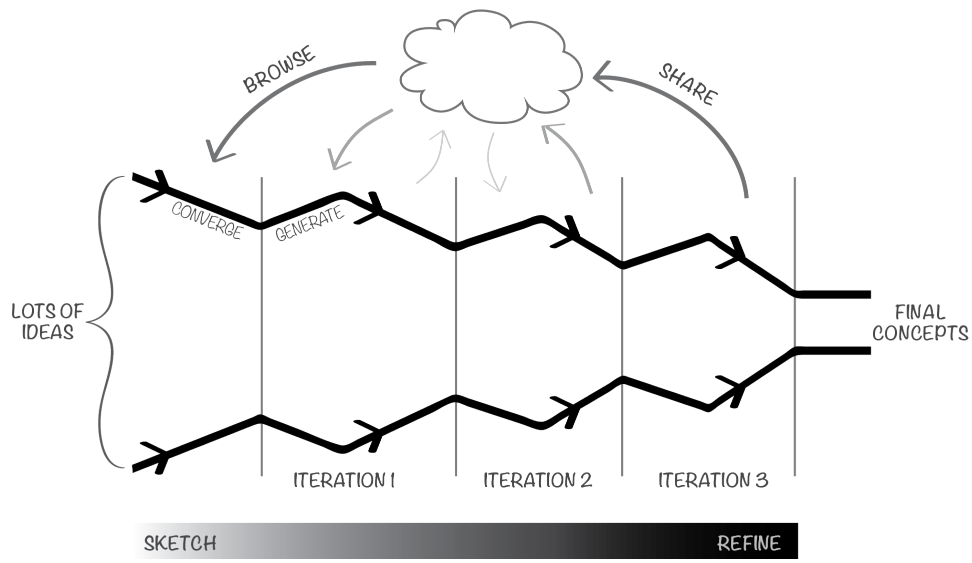


Figure 9.1: We found familiar design processes reflected in HaXD. We adapt the classic design funnel, where multiple initial ideas are iteratively developed [10], then add four design activities we have found useful when supporting design: *browsing*, *sketching*, *refining* and *sharing*. Exploring HaXD helped to reveal these activities, which are taken for granted in other fields but need to be explicitly supported for HaXD.

We believe scaffolding to be extremely important, as there are few haptic designers practicing today. Browsing is also tightly associated with the ability to Share designs and collaborate; when one designer shares their designs or ideas, others are able to browse it. We highlight three main challenges to supporting browsing in HaXD: representation, organization, and access.

Representations of single sensations: – How do we store, view, and organize haptic experiences? Haptic technologies are often inherently interactive, part of a multimodal experience with visual and audio feedback, and can take a variety of physical forms depending on the output (and input) device. This last point is particularly bothersome should the user not have access to the original device type – imagine trying to browse force-feedback sensations on your phone!

Collection classification and organization: – Haptic language and cultures of meaning are still in active development. Without a commonly shared lexicon, organization dimensions, or even adjectives, it is difficult to curate collections. Compare this to sound: most musical terms have a long tradition with a clearly defined lexicon (e.g., , crescendo, staccato); non-musical sound effects generally "sound like" something, and are often literal. With vision, one does not have to be a graphic designer or artist to instinctively understand "warm" and "cool" colours; the color wheel is introduced to us in grade school.

Overviews and skimming: – Collections of examples, especially visual or physical collections, are often displayed spatially for ambient reference or to enable quick scanning. When you can't feel multiple things, it can be hard to get the big picture or swiftly peruse a collection. Both designer and end-users have needs for finding similar/different vibrations in a collection, requiring a low barrier-to-entry on any overview technique. ⌈**OS** *also comment on search.*⌋

Sketch

Sketching allows people to form abstracted, partial views of a problem or design, iterate very rapidly and explore concepts. The generalized notion of sketching can support other activities: sketches are a notation that can be shared, and provide a vehicle for annotating and iterating on designs. Here, we use "Sketch" in contrast with Refinement, to refer to embodied exploration, concept generation, and initial design.

Early in our exploration, we found that our Haptic Instrument, mHIVE, excelled for exploring a design space but faltered for refining sensations. A real-time direct manipulation model in Tactile Animation facilitated both. In Macaron, we observed different levels of exploration and refinement, discovering a pattern of focused adjustment and repeated reflection [93], where the designers stepped back to feel their design in its entirety, and zoomed in to adjust parts in detail. We find this is a repeated pattern, where participants iteratively zoom in and out to different levels. The mixture of focus also depended on the stage of design - early on, exploration and execution take a more prominent role, while later refinement has smaller adjustments and more reflection. We have found this distinction useful when developing suites of tools, most explicitly when supporting design of CuddleBit behaviours. Voodle is a free-form Sketching tool, which can hand off to MacaronBit for Refinement. Future work remains to be done to establish whether there are discrete levels of focus, or if they lie along a continuum. In general, we've found two main features important for supporting Sketching: *abstraction and ambiguity*, and *rapid iteration*.

Abstractability and ambiguity: – Haptics suffers from a dearth of notation. Sketching of physical devices or interfaces is well supported, with paper and pencil and innumerable software assists. Sketching *motion*, and in particular showing what is or might be felt in, say, a vibrotactile experience, is trickier. While we can sketch a visual interface and look at it, it is much harder to sketch a haptic sensation and imagine it without feeling it.

Creative approaches are emerging. Most directly, Moussette and Banks [76] teach Haptic Sketching[1] with physical scraps and materials, combined with manual actuator and tools like Arduino, to build effective interactive haptic prototypes physically and programmatically in minutes or hours. Simple display-only sensations can be sketched (e.g., , VT icons) using interactive design tools [41**?**].

Abstractability and ambiguity?: –

Sketching and Refining – ⌈**OS** *What about a planning/execution/reflection pattern?*⌋

Refine

Clearly apparent in **??**, design requires iteration to refine an initial set of ideas into a single well-developed one through concept generation followed by iterative revision, problem-solving and evaluation, until only small tweaks are necessary. This long view of the design process is necessary to see designs through to the end; further, tweaking final designs is a valuable way to accommodate individual differences, and was adopted by our haptic designers in Chapter 8.

Incorporating haptic technology into a design is an extremely vertical process, dependent on specifics of hardware, firmware, software, application, and multimodal context. With the complexity of these many components, there can be a significant initial cost to setup a first haptic experience; then, adding this complexity to the time needed to program, recompile, or download to a microcontroller means iteration cycles have the potential to be slow and painful. Thus, increasing refinement fluidity is ripe for innovation. For example:

Pipelines – now connect initial design seamlessly through to final refinement [**? ?**]. Continuity in future tools will provide fluid, transparent (rather than cumbersome, many-staged) connection between hardware and software tools at different design stages.

Evaluation – is as crucial as for any human-centered refinement cycle. While it will often require some form of *sharing* (coming up next), here we simply point out that the full spectrum of evaluative mechanisms and supports found in user experience development can be gainfully applied to haptic design, from lab-based comparative performance studies to qualitative examination of how usage strategies change when a physical dimension is deployed (e.g., , [**?**]).

Customization tools – are appearing at least at level of prototyping and requirements generation [90, 94]. Force-feedback virtual environments support iteration and refinement through code, once the initial environment is setup. Software platforms like Unity[2] offer immediate control of variables in the UI itself.

[1] https://vimeo.com/29310105
[2] https://unity3d.com/

Context – – calibration, customization, and sensing – in tools will help final haptic designs remain consistent depending on user activity (e.g., , running impairs vibration sensitivity), individual differences, or other contextual concerns.

Share

Sharing designs is valuable at different stages of the design process [**?**], whether for informal feedback from friends and colleagues, formal evaluation when refining designs, or distributing to the target audience for use and community for re-use [**?**].

As haptic experiences must be felt, this process works best when collocated with only a few collaborators, whether by having collaborators work in the same lab, or by showing final experience in physical demos. During ideation, ideas can be generated when collaborating remotely, but physical devices need to be shipped back and forth and it is difficult to troubleshoot and confirm that configuration and physical setup are the exact same. Feedback also typically needs to be collocated, using in-lab studies or feedback, or shipping devices between collaborators. Furthermore, visual and audio design support very easy capture of ideas to share later, through smartphone cameras and microphones, that could later be browsed.

So far, haptic broadcasting, analogous to broadcasting radio or television (e.g., , Touch TV [74]) has been envisioned and explored. Followup work has added haptics to YouTube [**?**] and movies [54]. Low-cost devices like the HapKit [**?**][3] and Haply [32][4] make haptics more ubiquitous, but remain troublesome to calibrate. To share ideas remotely on phones, proxies like visualizations or other types of haptics (phone vibrations) could be used [**?**][5]. Features like automatic calibration and proxies for use in online evaluation, and online communities more generally, are still in development.

9.2.2 Generalizing Devices and Sensations

One major challenge facing haptic designers is the variety of haptic devices available. Each device has different physical properties, and may use different actuation principles or sensory modalities to communicate with the user. This might be analogous to how modern web sites employ responsive design, adapting to different screen sizes, albeit more extreme. A screen, at the end of the day, is that plane with a given physical size and pixel size; with haptics, contextual problems like grip can influence feedback, and haptic feedback can vary from single VT actuators to

[3] http://hapkit.stanford.edu/
[4] http://www.haply.co
[5] http://www.cs.ubc.ca/labs/spin/hapturk

VT grids, programmable friction, skin stretch, or force-feedback devices. We suggest three ways of managing this complexity: Grouping devices and interactions into *paradigms*, considering representational *translations*, and using affordances and closed-loop sensing to create *consistency*. A fourth related strategy, enabling customization, is so important we discuss it later.

Paradigms

However, we believe this problem is more diverse than screen size and resolution. We suggest this is a single *paradigm* for graphic designers, analogous to how haptic designers might work with several different types of single vibrotactile actuators. Haptic design becomes more diverse when designers need to consider arrays of VT actuators, programmable friction, skin stretch, or force-feedback, all of which are as analogous as screens are to, e.g., speakers.

First captured in Chapter 8, we believe that paradigms is a key concept to designing HaXD tools. We define a "paradigm" as an abstracted model of how to work (?) with a haptic device. Much like how different programming language paradigms, such as functional or object-oriented languages, can accomplish similar results but enable programmers to think in different ways more appropriate to their problem, we believe that different haptic or multimodal paradigms will enable different problem solving techniques for HaXD. In our in-depth studies we present three: an instrument paradigm, an animation paradigm, and a track-based editing paradigm.

It is critical to note that there is a many-to-many mapping between paradigms and haptic devices. Tactile animation can be used for multiple spatial grids and track based editors are generalized enough to handle multiple display types. Meanwhile, a multi-VT grid could be controlled by any of these three paradigms. As haptic displays become more diverse, we expect paradigms to play a larger role for organizing design perspectives, and multi-paradigm tools to become successful, just as multi-paradigm programming languages like Python and JavaScript afford flexibility, power, and accessibility - providing increasingly low barriers, wide walls, and high ceilings.

⌈**OS** *Paradigms can be split into temporal, spatial, and interactive (reactive) elements.*⌋

Representation and Native Platform

One striking problem with haptic technology is its sheer diversity. There are a wide diversity of devices, many of which support different paradigms, and all of which can be different based on their physical configuration. This poses a problem of access - if a designer creates a force-feedback sensation, how do you render that on a mobile device with a simple vibrotactile actuator? Can this translation be done?

Compare this to graphic design. At the end of the day, most graphic designs will be on a 2D plane, whether a screen or not. There are different sizes, resolutions, and colour maps - for example, print media might be designed differently than web - but similar tools and principles apply. In haptics, we might apply these different sizes and resolutions to a class of device - for example, a VT actuator can be an expensive C2 tactor or a low-cost voice coil. However, a single actuator is quite different from a 2D grid of actuators (like in Tactile Animation), and dramatically different from variable friction feedback (like with Rough Sketch) or 1-DoF force feedback (like with HandsOn). The question of resolution and platform within a single class of devices is analogous to the challenges faced by graphic designers and sound designers, but the diversity of classes of devices is even more varied.

Consistency through sensors and context control

A third way to adapt to the variety of contexts a haptic device might be used in is to either impose a known context, or sense and adapt an uncontrollable context.

Our industry haptic designers would talk about working with automotive companies, and how the material in the dashboard could affect the final haptic sensation. By controlling this material and working in a known environment, haptic designers might be able to keep their designs more consistent: When actuating a touch-screen in a car, a designer could know the materials. Similarly, wearable devices like the wristbands (Pebble, Apple Watch) know where the device will be worn, and can use that knowledge to their advantage; they can also use the materials of their wrist-straps to ensure a reliable tightness or pressure on the skin. Force-feedback devices like haptic knobs might change their handles, using physical affordances (Gibson? Norman?) to suggest a grip to the user.

Alternatively, closed-loop sensing might be able to standardize sensations. **?**] showed that accelerometers can provide insight into perceived loudness of VT stimuli. Techniques like activity classification (e.g., [**?**]), or even vibration sensing of a VT effect could help reduce noise from physical changes and material properties. This could be done *in-situ*, or during manufacturing for different device materials as a quality assurance step.

9.2.3 Designing Meaning

Because of the novelty of haptic design, we found that conceptual schemas are important for understanding the intent of haptic sensations. Schemas are existing conceptual models used as transitionary objects to understanding new concepts [85]. We found this procedure occurred not only in educational contexts, but also in design. In our early Haptic Instrument exploration, we found our participants' prior experience was a lens through which they interpreted haptic sensations. For example, one cat-owning participant interpreted likened sensations to cat purring, while another drew more on their knowledge of engines and cars. Heartbeats and rain [46] are effects that were easily understand by general participants, and verified using perceptual studies. Schemas are useful both for framing user interpretation of haptic experiences, and for designers themselves. We have also found the concept of design languages and narrative as helpful ideas when creating experiences.

Schemas as a tool for users

To interpret haptic signals, people employ a number of conceptual or translational schemas, often combining them. We might compare a haptic sensation to a natural one ("This is like a cat purring"), to emotions and feelings ("This is boring"'), or consider its potential usage (when a quickening tactile pulse sequence is described as a "speed up"). The meaning someone chooses is typically influenced by the sensation itself but also by the context of use and the user's background and past experiences [79**? ?**].

Facets are a concept originating from the domain of library and information retrieval which nicely capture the multiplicity and flexibility of users' sense-making schemas for haptic sensations. A facet is a set of related properties or labels that describe an aspect of an object [**?**]. Five descriptive facets have been proposed and examined for haptic vibrotactile stimuli [**?**]: physical properties, sensory properties, emotional connotations, metaphors, and usage examples.

If a designer neglects a consistent consideration of these meaning assignment facets the result is likely to be confusion and bad user experience. Leveraged properly, facet-driven mappings can be lead to more intuitive, consistent results and highlight pathways to work around individual differences, for example through tools that allow users to efficiently customize their interfaces.

Schemas can also be used for haptic designers to help frame their design processes. As covered in Chapter 2, many previous systems have their roots in other, non-haptic concepts, e.g., Touch TV []. This enables transfer effects; as we showed with Tactile Animation (Chapter 4), visual animators were able to easily create tactile designs.

Design Languages

Another way of framing a haptic experience is to use a *design language* like Google's Material Design (https://material.google.com). A design language is a defined set of aesthetic and interactive rules to ensure a consistent look-and-feel. We also believe that Gestalt-like principles will play a role in defining the possible design languages for haptics, much like musical concepts of thematic development, restatement, elaboration, and expectation, or graphical design concepts like contrast, repetition, alignment, and proximity are the tool with which languages can be formed. Previous efforts have identified frequency, amplitude, rhythm, affect, and spatial location as the main design elements for vibrotactile design. We have started to identify similar, middle level concepts - alignment and repetition emerged in the Macaron study.

Narrative Context and Multimodal Reinforcement

The intent of a haptic sensation was not always interpretable from the sensation itself. A great deal of the interpretation of an effect depended on the narrative context. Previous work used linguistic descriptions, such as "light rain" [46]. In our FeelCraft and Feel Messenger studies, we found that linking adaptable sensations to in-game stimuli was effective. With Feel Messenger, we at first attempted feel effects with abstract icons, such as "motor" for a rumble effect. In early piloting, these were not very helpful for understanding. Adding in vibrant cartoon emoji icons, and using a cat's face (for a purr) rather than a motor image were much more effective in sending haptic icons.

It is also worth mentioning the role of realism in a haptic experience. There are two types of realism, photo-realism (veridicality) and real-seeming (verisimillitude) (cite McCloud). For example, cartoons and animation are clearly not depictions of reality (they do not display veridicality), but they can appear life-like (they display verisimilitude). In this dissertation, we focused on verisimillitude rather than veridicality; the latter is often already accommodated by realistic rendering techniques. We believe both require a design process, but more photo-realistic approaches have different challenges than cartoony ones, and may require more of an engineering perspective than a design perspective. That said, sometimes combining the two is helpful; much like how in computer graphics, physical simulation and realistic rendering are used as a baseline, but artists still may want artistic control over certain scenes or effects.

9.2.4 Customization

We've shown how individual differences are a prominent feature of haptic perception and psychology. Furthermore, variability in and poor designer control over context – user attention and device form factor and manner of connection, as well as use environment – mean that haptic sensations often need to be tuned to both each person and each use case.

The most likely solution is to make it easy for end-users in populations that might benefit from a haptic modality to *customize* aspects of haptic design elements, whether by choosing pre-formed settings and "skins", adapting defaults, or wholly designing their own. Possible approaches range from volume-like slider controls, options to select sensations from curated collections, or, at the more complex end, perceptually-confirmed filters like those found in Instagram or PhotoShop [90, 94**?**].

Volume Controls – As found in our interviews with designers, the end-user context is sometimes unknowable. There is not much designers can do other than putting a volume control on their design to facillitate customization. In informal discussions with engineers from commercial companies (e.g., D-Box), volume controls were a critical, early addition to their haptic chairs. As explored in our FeelCraft and FeelChat projects, the extra level of volume control is implementable, expressive, reduces footprints, but is more limiting for designers and requires programming expertise. While there is a great deal of noise in perceptual parameters for high-level controls, careful investigation can improve this [46, 94].

Novelty and Irritation – Haptic sensations are new, giving them great power and great responsibility. They can grab attention, but designers must take care not to annoy with constant haptic feedback. The balance is tricky. Good design might be not even noticed when present. With RoughSketch, we found that allowing a button to remove haptic feedback was very persuasive for appropriate, non-irritating designs. We also found that muting features were essential for HaXD design tools.

9.3 HaXD Tools: Designing and Implementing

In this section, we comment on the specific features and forms HaXD support tools might involve. Because of the diverse activities described in the previous section, we believe there is no silver bullet: haptic design tools are likely to form a suite or ecosystem. Here, we discuss ways to enable creativity through mature tools with "a low barrier, wide walls, and a high ceiling." We talk about important collaborative techniques and how to deploy implemented tools. We discuss implications for developers and software engineering teams. Finally, we conclude with some comments on our evaluation methodology and how future HaXD tools might be evaluated.

9.3.1 Communities and On-line Deployment

Unfortunately, haptic technology faces obstacles to Browsing and Sharing, especially when under development; this has typically confined distribution and exposure to lab prototypes available only to in-person visitors. Demos are difficult to setup and there is little infrastructure in place for distribution. While synchronous, collocated collaboration is effective with demos, remote or asynchronous collaboration is rife with trouble. Consistency must be maintained, and setup can be painful for those not trained in setting up devices. On top of this, we found a strong tendency to show, not tell sensations, and an impoverished language especially for non-experts. Without a shared reference point of physically feeling the sensation, we believe that design can be easier, especially when a language of design is underdeveloped.

Recently, online interfaces have emerged for sourcing content, distributing content, and distributing media itself. We found that moving to a web-based tool with Macaron greatly facillitated distribution. Online deployment simultaneously widens exposure and speeds development, making it easier for designers to be inspired by or directly build upon one another's work. The trend will accelerate as the field matures, but there will need to be concurrent development of accessible hardware to connect with software tools.

Crowdsourcing and broadcasting:

We reviewed some of the substantial challenges and spoke of one type of solution. In HapTurk (Chapter 6), we showed proxies of high quality haptic experiences can communicate key aspects on more shareable media, to gain access to crowdsource evaluation tools like Mechanical Turk. Proxies might be able to generalize in other ways, for example, using video to infer feedback about physically moving objects like the CuddleBits (e.g., as in [**?**]).

Proxies can do more than elicit feedback from crowds. It might also be a viable way to translate sensations between representations, analogous to downsampling high-definition video broadcasts for standard definition televisions. More exotic proxies, like using vibrotactile feedback to represent force or friction could be considered. Future perceptual studies are needed to accomplish this.

Open Haptics – design sharing communities:

Other kinds of designer-facing online communities and outreach activities may assist with open haptic media – making it easier to share design resources, build up a haptics design culture, and, where possible, cooperate on establishing a consistent design language of haptics. For example, online software like VibViz [**?**] and Macaron (Chapter 5) provide details to designers anywhere, while open hardware projects like HapKit [**?**] and Haply [32] are available to hackers and students. Conference workshops and hardware kits provide users and designers with additional means to experiment with advanced haptics.

Each of these projects solve different problems and provide independent benefits. Online collaboration, and resources like articles on haptic perception or tutorials on how to effectively create haptics, will connect more designers, artists, developers, students, and hackers and help to build haptics into new user experiences.

Examples, demos, Show, don't tell

We found that demonstrations were critical for haptic designers in the wild. To help elicit requirements, customers were brought in to try out various demos. To persuade customers, actually feeling the haptics was critical, if not always enough. The Haptic Instrument showed evidence for deictic features - what do you think of "this", which facillitated direct demonstration without having to resort to indirect linguistic descriptions. In Macaron, exposing the underlying structure of examples lead to improved understanding of how to develop sensations and developed haptic idioms.

Risk – Adding haptics is risky. ⌈**OS** *we say this quite well in the previous 8.4 discussion, why should we repeat it?*⌋

9.3.2 Towards a Mature HaXD Tool Suite

Soft Features – Repeatedly, participants asked for "soft features", associated with more polished tools. This includes features like copy and paste, undo and redo, saving and loading, grouping, looping, reduced delay, and high-fidelity rendering. We increasingly found, as we iteratively developed HaXD tools, that these features are more important than getting the right paradigm. As long as a designer is able to freely and accurately create, they can work around awkward design metaphors.

Low barrier, wide walls, high ceiling – A major goal of our work was to support creativity with haptic technology, and to support low barriers, wide walls, and a high ceiling [86**?**]. We've confirmed that this is critical for HaXD, and found ways of accomplishing it. As we mentioned, our participants found soft features essential, which helped to free them to make mistakes, explore various options, and provide both accessibility (low barriers) and refinement (high ceilings). For example, in Tactile Animation, participants found the animation objects easier for exploration and non-expert designers, while vector sensations offered more fine-tuned control. Similarly, we found the more flexible track-based paradigm used by Macaron to allow for various paths through the interface, and was generalizable to other sensations like simple 1-DOF robots.

Making haptics means different things at different stages in the process (Section **??**). Media creation in other modalities is supported with a wealth of tool specializations that recognize these diverse needs. Haptic making has reached a maturity that demands tool power, nuance, and specialization as well.

Some of the important ground to cover here includes better support for *sketching*, *high-level manipulations*, *multimodal qualitative analysis* and *browsing*, as well as workflow integration and addition of specific useful features to tools that already do exist.

Allowing Sketchiness

Early design needs support for low-cost, rapid ideation that elevates key points without extraneous detail. Haptic design presents a few challenges; solving these will further empower designers.

Ambiguous sketches show only necessary detail or relevant view points, helping a designer work with half-formed ideas and questions, but a haptic sensation requires a physical implementation which must be exactly specified. We need to explore different approaches to quick prototyping of different haptic aspects – e.g., , feel, form factor, timing. One approach is *modularity*: prototyping one aspect at a time, then integrating into more expensive engineering prototypes once the design space has been narrowed. Here, the tool need is both for ready-to-go platforms to explore single-aspect ideas, and ways to efficiently extract, port, or integrate best-of-breed sketches into later design stages.

Good defaults: – In another approach, designers will be able to select from carefully chosen default settings, and/or specify any known constraints with other details be filled in automatically.

Annotation: – Sketching should supply markup support for both early-stage designers themselves and the stakeholders they share with, to annotate sketches, circle problem areas, and write down related ideas.

Ad-hoc use: – Finally, tools should be easy to access. With a pen, any napkin is a canvas. Haptics (and multimodal interactions in general) need napkins too.

Fluid control of low-level parameters

A viable approach enabled by our real-time tools is an implicit understanding of the sensations by designers, who were able to work with low-level engineering parameters like frequency, amplitude, or even voltage. In the Tactile Animation study, we found that while animation objects were more used and described are more accessible for non-animators, vector sensations (directly controlling each actuator) were described as useful for finer, expert control. Learning to control low-level parameters is important for "high ceilings" in HaXD support tools.

High-Level Manipulations

The haptics community currently has access to a number of editing tools which together exhibit a variety of approaches to editing and authoring, primarily of low level effect detail (Section **??**). However, being able to manipulate sensations in the large or in an expressive way will enable designers to access a larger design space, and do more in less time.

In other modalities (Photoshop for graphic media, Audacity for sound), this kind of functionality appears in a variety of ways. *Filters* are essentially "tuners" that move a sensation along dimensions of direct design interest. *Transformations* allow users to alter or distort a segment, e.g., , to darken or brighten it, stretch its time base in a linear or nonlinear way, or shift component balance. *Large-scale manipulations* can move, combine, subtract and otherwise alter design elements.

The scope of such manipulations should span perceptual, informational, and affective perspectives. Research is underway to understanding these manipulations from user perceptual and subjective standpoints, and implement them algorithmically.

Engineering useful features into mature tools

Both individual experiences in designing with haptics, and reference to tools in other modalities show the value of seamless access to many small but useful features – direct manipulation, undo/redo, copy/paste, selection and group manipulation, import/export to various formats, access to a library, etc.. Individually, many of these do not present major research or usability problems, but integrating them is another matter. There are at least two obvious approaches:

Additive: – Haptics authoring capability can be added to existing mature platforms focused on another modality – e.g., , to Photoshop (for graphics) or Premier (video). Force-feedback is already integrable into certain video game and virtual world environments (Unity, XNA), but this is only a small subset of possible haptic experiences.

Haptics-specific: – To truly optimize haptics capabilities, it may be worthwhile to invest serious development into a haptics-specific platform; or more likely, many attempts at such a thing focused on different categories of hardware (e.g., , tactile wearables versus 3D desktop force feedback environments) or application area, perhaps by extending initial low-level editors already available.

Researchers are good at pioneering individual steps and features, but integration is a significant development initiative which lies more in the province of businesses who stand to profit by the effort invested. Which task focus and development pipelines are to be supported, including multimodal design and integration, will be driven by application areas with urgent needs and a promising economic prognosis: e.g., , for movie special effects versus surgical simulations or end-user customizations.

Iteration

In our interviews, we found iteration was extremely painful. Despite this, we've found iteration is necessary (as with all design processes), but even more so because it is difficult to understand requirements or identify a good design without feeling it. Many of our designers would iterate until it "felt right". Thus, iteration is very necessary yet very painful.

9.3.3 Notes on Implementing HaXD Tools

During our in-depth studies, we looked at three different platforms for implementing HaXD support tools: Android on tablet for mHIVE, Python on desktop for Mango, and JavaScript deployed on the web for Macaron. From these three implementations, and from attempting to distribute or deploy them, we have found several lessons for implementing interactive software for HaXD support. Many of these are straightforward applications of software engineering, but we particularly found recently developed web tools to be especially useful when managing the complex interfaces required by design tools.

Observer Pattern – Well established as a design pattern [], the found the observer pattern was critical in keeping the complex interfaces of our design tools synchronized. This is especially critical because we need to synchronize haptic, audio, and video playback, and allow implementations through both temporal and spatial means. In Mango, we implemented this pattern ourselves, but in Macaron, we used React [] which automates the process. This latter approach was more efficient and facillitated programming.

Components and Track-based Editing – Another valuable approach to develop extensible tools was React's Components, composable interface elements that can be parameterized and reused. This technique, along with a carefully organized architecture and the generic track-based editor control paradigm used by Macaron made it very easy to extend. When developing MacaronBit by forking the original code for Macaron, we were able to repurpose the tool to control the CuddleBits in a single day. While a better paradigm could be designed, our in-lab experts were able to used generalized tracks to create differentiable behaviours from the bit. One can also imagine redesigning a track-based editor to use a different control paradigm with the same device - we do this with MacaronMix, which mixes two VT icons using again a fork off of Macaron.

Flexible, extensible data structures – Because of the complexity and variance of haptic devices, we needed to establish data structures that were lightweight but extensible. We used JavaScript Object Notation (JSON) to accomplish this in Mango, Macaron, and FeelCraft. JSON which allowed us to specify only the required information needed, whether for device parameters in Mango, or sensations created by both Mango and Macaron. Our tools would check JSON files for designed features - like amplitude tracks in Mango - and ignore others we left in for a stub - like frequency tracks. These structures were simple, human-readable, ignored extraneous detail, and could gracefully fail when poorly formatted. Because we need to accommodate different paradigms in HaXD tools, lightweight data types make a lot of sense.

Existing data structures like .WAV – Audio control useful, Arduino connection useful - but how to connect to web?

Web Deployment – One challenge with haptic technology Web - more people, online communities, resources, show internal structure Need to package up software Documentation for use

No Delay – In interviews with designers, we found that delay was critical; multimodal sensations had to be tightly synchronized. We found similar problems with our tools. mHIVE had a short delay which was distracting to participants, particularly when they tried to play very short pulses. Macaron's real-time audio synthesizer sometimes created a "muddy" effect (P?) because it had a limited update rate. We found this disappeared after implemented .WAV exporting function. Combined with our findings that users would switch between focused, in-depth editing and more reflective experiencing, we suggest (and are working to implement) a pre-feel/render model, like with visual effects tools. This is one potential problem with current web implementations of our design tools, especially when attempting to communicate to embodied devices like Arduino.

Distribution requires hardware – An obvious, but critical, requirement for HaXD tools is that they require the appropriate hardware. Above we've discussed flexible architectures and data structures to support different hardwares and paradigms of sensations. In addition, potential designers need easy access to hardware. One way we've found is to create do-it-yourself haptic wristbands for Macaron. (URL) Critical for VT feedback is that amplifiers and actuators matter - we found the DIY custom amplifier we've built does not have the same crispness as a more expensive commercial audio amplifier. This means that modular systems could be designed for facillitate accessible haptic feedback and design, but are complicated in because the setup doesn't end at the actuator. Another approach is to use a HapTurk-based translation with mobile vibrations like those available on Android (link to hapticJS), although this impoverishes sensations and leads to the challenge of native platform, discussed above.

Logging and Analytics – One important feature for modern software is remote analytics. Various types of metrics and usage statistics help developers prioritize development, fix bugs, and report to investors. Interestingly, because haptic technology extends into the physical world, one can consider using remote analytics both on their design and playback software, and on the physical devices. With a modular system Logging and analytics not a problem; should record these features (from Macaron, a little Tactile Animation)

9.3.4 Evaluating HaXD Tools

Flow, Learning, Scaffolding, and Embodiment

Exploratory Analysis and Visualization Tools (Haptic Analytics)

Haptic researchers frequently analyze haptic datasets with the goal of informing next steps in a design use case or deriving design guidelines for the haptic community. Such an investigation usually involves exploratory analysis of quantitative and qualitative attributes of signals and users' perceptions. Currently, designers use a potpourri of general-purpose software (e.g., , Matlab, R, SPSS, Tableau) for their analysis. Each tool provides just a partial view of the data; the difficutly of integrating their insights hinders the analysis process.

Visual analytics interfaces, which support analytical reasoning through interactive visualization of datasets, are largely absent from haptics at this time. Examples of desirable functionalities include easy access to the feel and source of signals, enabling rapid and flexible organization of haptic signals according to their various properties, allowing researchers (and the crowd) to attach metadata (tags, genealogy, annotations) to elements and subsets of a haptic collection, and do analytics on this metadata.

The first steps towards haptic analytic interfaces (Section **??**) have exposed some of the work that's needed to make these truly useful. Much of it is the same underlying perceptual knowledge and algorithmic advances required for authoring and manipulation, e.g., , of perceptual dimensions and user's mental organization and ways of differentiating and interpreting sensations, and this is well underway. Connection to or integration of multimodal functionality – e.g., , suitability for coordinating with design elements from other modalities, in various communication roles (Section **??**) – will develop along with our experience in working with these modalities.

Post interviews – Early in our investigation, we found that participants found traditional think-aloud [] protocols challenging. This appeared to be due to a split attention of paying close attention to sensory input on their hand while simultaneously designing sensations. Even without thinking-aloud, this was cognitively challenging, and participants wanted looping, or would step back to refine and feel their sensations. In response, we adopted unintrusive logging and post-interviews, both of which were effective in capturing aspects of the haptic design process.

Phenomenology – Because of the close connection with participant experience, we adopted techniques from phenomenology, both to capture the sensory experience of haptic sensations, and to capture the experience of design. This perspective was useful for rigorous examination of our participants' interviews and to guide the interview process. In addition, the practical techniques, considering each meaning unit and clustering them into both in-vivo language and analytical language, very useful, especially when later combined with Grounded Theory techniques.

Grounded Theory – To better equip ourselves to analyze larger sets of data and develop a broader theory, we also drew upon Grounded Theory as described by Corbin and Strauss []. Especially when analyzing screen-captured video, coding techniques were valuable in identifying countable data, strategies, and adding a second layer of data analysis techniques to complement those provided by phenomenology.

Analytics and Logs – Qualitative analysis of post-interviews provided valuable insight, but only gave a partial view of haptic experience design. Timing analysis, and visualized logs, captured data that may not have been not have been observed by the researcher, and not reported or noticed by the participant in a post-interview. Importantly, it also offers a transition into broader data collection at large, providing a sustainable method of observing features

Multimodal Design Tasks – Throughout our three in-depth design tool studies, we refined our approach for participant tasks. With the initial Haptic Instrument study, we asked participants to freely explore the interface and attempt to design for affective words like happy or sad. This was challenging for the participants, and we found inconsistent designs. ⌈**OS** *TODO: what exactly was bad about this?*⌋ In our Tactile Animation study, we used more defined scenarios - a heartbeat, a directional cue for driving, and a sound-based design task. The first two were effective and descriptive for our participants, but the sound task in particular engaged them; it also gave us controls over theoretical dimensions like complexity or abstractness, impose constraints like careful timing, and was externally valid with the multimodal nature of haptic design. In our final in-depth study, we adopted animations instead of sounds, with similar results.

Ratings – In our studies, we had participants rate their designs including confidence and difficulty. We did not find quantitative results from these reports, mostly due to low sample size and the qualitative nature of analysis. However, we did find it to be useful for two reasons. First, it provided an elicitation device, inviting participants to comment on the difficulty or their happiness with their designs. Second, as we scale to more natural settings and higher sample sizes, we hope quantitative findings from rating scales will provide insight to questions about challenge, learning, and flow theory, which emerged from the third in-depth study.

9.4 Current and Future Work

⌈**OS** *todo*⌋

9.5 Conclusion

⌈**OS** *todo*⌋

9.6 Previous HaXD theory

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 9.2). 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* [14], 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.

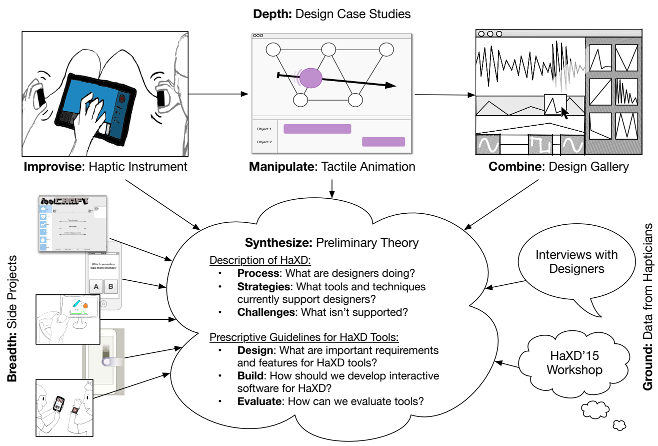


Figure 9.2: Planned synthesis of data for a preliminary theory of haptic experience design.

9.7 Data Collection

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

9.7.1 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.

9.7.2 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.

9.7.3 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.

9.8 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.

9.8.1 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.

9.8.2 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.

9.9 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 [50]), 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.

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 11, 35, 38

[2] Android Open Source Project. Android Developers, 2012. URL http://developer.android.com/index.html. → pages 23

[3] O. Bau, I. Poupyrev, A. Israr, and C. Harrison. TeslaTouch: Electrovibration for Touch Surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10*, page 283, New York, New York, USA, oct 2010. ACM Press. ISBN 9781450302715. doi:10.1145/1866029.1866074. URL http://dl.acm.org/citation.cfm?id=1866029.1866074. → pages 11

[4] R. A. Bolt. Put-that-there: Voice and Gesture at the Graphics Interface. *ACM SIGGRAPH Computer Graphics*, 14(3):262–270, July 1980. ISSN 00978930. doi:10.1145/965105.807503. URL http://dl.acm.org/citation.cfm?id=965105.807503. → pages 26

[5] 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 17, 23

[6] 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 17

[7] 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 17

[8] 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 17, 23

[9] 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 17, 23

[10] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007. → pages xvii, 17, 18, 47, 48, 105, 106, 138, 139

[11] 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 17

[12] 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 14, 17, 66, 67

[13] A. Chang and C. O'Sullivan. Audio-haptic feedback in mobile phones. In *CHI EA '05*, pages 1264–1267, Apr. 2005. ISBN 1595930027. doi:10.1145/1056808.1056892. URL http://dl.acm.org/citation.cfm?id=1056808.1056892. → pages 14

[14] J. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc., 3 edition, 2008. → pages 20, 25, 41, 52, 98, 104, 109, 110, 156

[15] J. W. Creswell. *Qualitative inquiry and research design: choosing among five approaches*. Sage Publications Inc, 3rd edition, 2013. → pages 19, 26

[16] N. Cross. *Designerly Ways of Knowing*. Springer-Verlag London Limited, 2006. → pages 18, 106

[17] N. Cross. Creative thinking in design. In *Proceedings of the 2007 Symposium on Science of Design - SoD '07*, page 2, New York, New York, USA, Mar. 2007. ACM Press. ISBN 9781605584362. doi:10.1145/1496630.1496632. URL http://dl.acm.org/citation.cfm?id=1496630.1496632. → pages 20

[18] N. Cross. *Design Thinking: Understanding How Designers Think and Work*. Berg Publishers, Oxford, UK, 2011. → pages 20, 29

[19] 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 15, 33

[20] 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 10, 16, 48, 107

[21] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, A. Lécuyer, and M. Christie. HapSeat: producing motion sensation with multiple force-feedback devices embedded in a seat. In *VRST '12*, pages 69–76, New York, New York, USA, Dec. 2012. ACM Press. ISBN 9781450314695. doi:10.1145/2407336.2407350. URL http://dl.acm.org/citation.cfm?id=2407336.2407350. → pages 14, 44

[22] F. Danieau, A. Lecuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie. Enhancing audiovisual experience with haptic feedback: a survey on HAV. *IEEE transactions on haptics*, 6(2):193–205, jan 2013. ISSN 2329-4051. doi:10.1109/TOH.2012.70. URL http://www.ncbi.nlm.nih.gov/pubmed/24808303. → pages 14

[23] 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 14, 16, 108

[24] 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 18, 20, 47, 59, 106

[25] 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 17

[26] P. Ekman, I. R. Davidson, P. Ellsworth, and W. V. Friesen. Are There Basic Emotions? *Psychological Review*, 99(3):550–553, 1992. → pages 25

[27] 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 18, 106

[28] 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 12, 15, 17, 29, 33, 36, 48, 108, 109

[29] 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 17

[30] FMOD. FMOD Ex API, 2013. URL http://www.fmod.org. → pages 23

[31] 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 12, 16, 108

[32] C. Gallacher, A. Mohtat, and S. Ding. Toward Open-Source Portable Haptic Displays with Visual-Force-Tactile Feedback Colocation. In *HAPTICS '16*, 2016. ISBN 9781509009039. → pages 12, 142, 148

[33] Y. Gao, H. A. Osman, and A. El Saddik. MPEG-V based web haptic authoring tool. In *IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*, pages 87–91. IEEE, oct 2013. ISBN 978-1-4799-0849-3. doi:10.1109/HAVE.2013.6679616. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6679616. → pages 14

[34] 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 16, 108

[35] 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 13, 15, 17, 23, 108

[36] 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 18, 106

[37] 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 11, 16, 33, 108

[38] V. Hayward. Tactile Illusions. In *Scholarpedia of Touch*, pages 327–342. Atlantis Press, Paris, 2016. doi:10.2991/978-94-6239-133-8_27. URL http://link.springer.com/10.2991/978-94-6239-133-8{_}27. → pages 10, 11

[39] 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 9, 16, 106, 108

[40] 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 17, 18, 33, 47, 48, 105, 106

[41] 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 15, 108, 140

[42] 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 16, 18, 105

[43] W. IJsselsteijn. Presence in the past: What can we learn from media history? In *Being There - Concepts, Effects and Measurements of User Presence in Synthetic Environments*, pages 17–40. IOS Press, Amsterdam, 2003. → pages 13

[44] 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 11, 32, 35, 37, 38, 59

[45] A. Israr, S.-C. Kim, J. Stec, and I. Poupyrev. Surround haptics: tactile feedback for immersive gaming experiences. In *CHI EA '12*, pages 1087–1090, New York, New York, USA, May 2012. ACM Press. ISBN 9781450310161. doi:10.1145/2212776.2212392. URL http://dl.acm.org/citation.cfm?id=2212776.2212392. → pages 10

[46] 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 16, 17, 48, 82, 107, 144, 146, 147

[47] 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 67, 68, 85

[48] 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 17, 29, 109

[49] J. Johnson and A. Henderson. Conceptual models: begin by designing what to design. *Interactions*, 9(1):25–32, Jan. 2002. ISSN 10725520. doi:10.1145/503355.503366. URL http://dl.acm.org/ft_gateway.cfm?id=503366&type=html. → pages 33

[50] L. Jones. News from the Field: Courses in Haptics. *Transactions on Haptics*, 7(4):413–414, 2014. → pages 12, 16, 108, 160

[51] L. Jones, M. Nakamura, and B. Lockyer. Development of a tactile vest. In *HAPTICS '04*, pages 82–89. IEEE, 2004. ISBN 0-7695-2112-6. doi:10.1109/HAPTIC.2004.1287181. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1287181. → pages 10

[52] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000. → pages 12, 107

[53] M. Kim, S. Lee, and S. Choi. Saliency-driven real-time video-to-tactile translation. *IEEE transactions on haptics*, 7(3):394–404, jan 2014. ISSN 2329-4051. doi:10.1109/TOH.2013.58. URL http://www.ncbi.nlm.nih.gov/pubmed/25248221. → pages 14, 45

[54] 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 10, 14, 15, 17, 32, 33, 108, 142

[55] K. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):219–230, mar 2006. ISSN 1077-2626. doi:10.1109/TVCG.2006.32. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1580456. → pages 12

[56] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, jul 1987. ISSN 00100285. doi:10.1016/0010-0285(87)90008-9. URL http://www.sciencedirect.com/science/article/pii/0010028587900089. → pages 9

[57] S. J. Lederman and R. L. Klatzky. Haptic perception: A tutorial. *Attention, Perception & Psychophysics*, 71(7):1439–1459, oct 2009. ISSN 1943-3921. doi:10.3758/APP.71.7.1439. URL http://www.springerlink.com/index/10.3758/APP.71.7.1439. → pages 9

[58] 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 16, 108

[59] 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 20, 47, 48, 49, 58

[60] 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 15, 17, 29, 33, 108

[61] J. Lee and S. Choi. Real-time perception-level translation from audio signals to vibrotactile effects. In *CHI '13*, pages 2567–2576, Paris, France, Apr. 2013. ISBN 9781450318990. doi:10.1145/2470654.2481354. URL http://dl.acm.org/citation.cfm?id=2470654.2481354. → pages 14, 33

[62] 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 15, 17, 22, 29, 108

[63] J. Lee, Y. Kim, and G. Kim. Funneling and saltation effects for tactile interaction with virtual objects. In *CHI '12*, page 3141, New York, New

York, USA, May 2012. ACM Press. ISBN 9781450310154. doi:10.1145/2207676.2208729. URL http://dl.acm.org/citation.cfm?id=2207676.2208729. → pages 11, 35

[64] V. Levesque, L. Oram, K. MacLean, A. Cockburn, N. D. Marchuk, D. Johnson, J. E. Colgate, and M. A. Peshkin. Enhancing physicality in touch interaction with programmable friction. In *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*, pages 2481–2490, New York, USA, May 2011. ACM Press. ISBN 9781450302289. doi:10.1145/1978942.1979306. URL http://dl.acm.org/citation.cfm?id=1978942.1979306. → pages 11, 103

[65] P. Lopes, A. Ion, and P. Baudisch. Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pages 11–19, New York, New York, USA, 2015. ACM Press. ISBN 9781450337793. doi:10.1145/2807442.2807443. URL http://dl.acm.org/citation.cfm?doid=2807442.2807443. → pages 13

[66] J. Luk, J. Pasquero, S. Little, K. MacLean, V. Levesque, and V. Hayward. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 171, New York, New York, USA, apr 2006. ACM Press. ISBN 1595933727. doi:10.1145/1124772.1124800. URL http://dl.acm.org/citation.cfm?id=1124772.1124800. → pages 11

[67] 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 17, 134

[68] 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 16, 108

[69] 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 16, 108

[70] K. E. MacLean, M. J. Enriquez, and T. Lim. Morphing in periodic tactile signals. In *World Haptics 2009*, pages 178–183. IEEE, 2009. ISBN 978-1-4244-3858-7. doi:10.1109/WHC.2009.4810844. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4810844. → pages 12

[71] 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 47, 48, 49

[72] T. H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Chicago, IL, 1994. → pages 12

[73] D. J. Meyer, M. A. Peshkin, and J. E. Colgate. Fingertip friction modulation due to electrostatic attraction. In *2013 World Haptics Conference (WHC)*, pages 43–48. IEEE, apr 2013. ISBN 978-1-4799-0088-6. doi:10.1109/WHC.2013.6548382. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6548382. → pages 11

[74] S. O. Modhrain and I. Oakley. Touch TV : Adding Feeling to Broadcast Media. 2001. → pages 13, 17, 142

[75] 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 16, 106, 108

[76] 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 16, 29, 33, 106, 108, 140

[77] C. Moustakas. *Phenomenological Research Methods*. SAGE Publications, Inc., 1994. ISBN 978-0-8039-5798-5. → pages 19, 20, 24, 41, 52, 109, 110

[78] 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 16

[79] 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 17, 19, 29, 109, 145

[80] 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 17, 109

[81] A. Okamura, J. Dennerlein, and R. Howe. Vibration feedback models for virtual environments. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1, pages 674–679. IEEE, 1998. ISBN 0-7803-4300-X. doi:10.1109/ROBOT.1998.677050. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=677050. → pages 9

[82] A. M. Okamura. Haptic feedback in robot-assisted minimally invasive surgery. *Current Opinion in Urology*, 19(1):102–107, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19057225. → pages 12

[83] A. M. Okamura, S. Chan, B. Hannaford, K. MacLean, and W. Provancher. Best Practices for Teaching Haptics, 2012. → pages 16, 108

[84] 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 10, 15, 33, 108

[85] 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 13, 144

[86] 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 18, 33, 106, 149

[87] P. Richer. A phenomenological analysis of the perception of geometric illusions. *Journal of Phenomenological Psychology*, 8(2):123–135, 1978. → pages 19

[88] 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 23

[89] 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 15, 33, 48, 108

[90] O. Schneider, S. Zhao, and A. Israr. FeelCraft: User-Crafted Tactile Content. In *Proceedings of AsiaHaptics 2014*, 2014. → pages iv, 48, 103, 107, 141, 146

[91] O. S. Schneider and K. E. MacLean. Improvising Design with a Haptic Instrument. In *HAPTICS '14*, Houston, USA, 2014. → pages iii, 21, 33, 63, 103, 108

[92] 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 iii, 21

[93] D. A. Schön. *The Reflective Practitioner*. 1982. → pages 17, 18, 20, 48, 105, 106, 138, 139, 140

[94] 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 17, 48, 109, 141, 146, 147

[95] H. Seifi, K. Zhang, and K. MacLean. VibViz: Organizing, visualizing and navigating vibration libraries. In *World Haptics '15*, 2015. → pages 17, 48, 49, 50, 58, 64, 65, 66, 77, 103, 107, 109

[96] 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 11

[97] M. J. Shaver. *The Twiddler: A Haptic Teaching Tool: Low-Cost Communication and Mechanical Design*. PhD thesis, UBC, 2003. → pages 12

[98] 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 17, 18, 105, 106

[99] 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 15, 33, 48, 59, 108

[100] 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 15, 33, 59, 108

[101] H. Tan, A. Lim, and R. Traylor. A psychophysical study of sensory saltation with an open response paradigm. *nternational Symposium on*

Haptic Interfaces for Virtual Environment and Teleoperator Systems, 69: 1109–1115, 2009. → pages 11

[102] 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 17, 68

[103] D. Tsetserukou, A. Neviarouskaya, H. Prendinger, N. Kawakami, and S. Tachi. Affective haptics in emotional communication. In *Proc. ACII '09*, pages 1–6, Sept. 2009. doi:10.1109/ACII.2009.5349516. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5349516. → pages 15

[104] O. A. J. Van Der Meijden and M. P. Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical Endoscopy*, 23(6):1180–1190, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19118414. → pages 12

[105] R. T. Verrillo and G. A. Gescheider. Perception via the sense of touch. *Tactile aids for the hearing impaired*, pages 1–36, 1992. → pages 38

[106] 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 17, 18, 47, 48, 53, 59, 105, 106, 138

[107] J. Watanabe, T. Hayakawa, S. Matsui, A. Kano, Y. Shimizu, and M. Sakamoto. Visualization of Tactile Material Relationships Using Sound Symbolic Words. In P. Isokoski and J. Springare, editors, *Haptics: Perception, Devices, Mobility, and Communication*, volume 7283 of *Lecture Notes in Computer Science*, pages 175–180. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-31403-2. doi:10.1007/978-3-642-31404-9. URL http://www.springerlink.com/index/10.1007/978-3-642-31404-9. → pages 29

[108] G. Wilson, T. Carter, S. Subramanian, and S. A. Brewster. Perception of ultrasonic haptic feedback on the hand. In *CHI '14*, pages 1133–1142, New York, New York, USA, Apr. 2014. ACM Press. ISBN 9781450324731. doi:10.1145/2556288.2557033. URL http://dl.acm.org/citation.cfm?id=2611105.2557033. → pages 44

[109] L. Winfield, J. Glassmire, J. E. Colgate, and M. Peshkin. T-PaD: Tactile Pattern Display through Variable Friction Reduction. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 421–426. IEEE, mar 2007. ISBN 0-7695-2738-8. doi:10.1109/WHC.2007.105. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4145211. → pages 11

[110] 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 10, 23, 107, 134

[111] 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 10, 107

[112] 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 17, 29, 109