

Designing a Haptic Animation Tool

Author(s) withheld for anonymous submission

Affiliation(s)

Address(es)

e-mail address(es)

ABSTRACT

Despite the rapid proliferation of spatial haptic devices, such as actuated chairs and wearables, we are still limited in our means of designing their output. Existing tools either control single actuators, limiting their application space, or control multiple actuators individually, splitting a single sensation into independent and difficult-to-control tracks. We introduce Mango, a haptic animation tool that facilitates the creation of haptic sensations as a single spatio-temporal percept rather than a set of actuator tracks. By using an animation metaphor and generalized rendering algorithm, we open the way for visual animators to transfer their expertise, designing haptics in a similar fashion. We describe Mango's design, including requirements drawn from interviews with designers, previous haptic control interfaces, and visual animation interfaces. Two studies psychophysically optimized a rendering algorithm for arbitrary spatial arrays of voice coils, and validated the animation metaphor with animator interviews.

Author Keywords

Haptics; vibrotactile; animation; design.

ACM Classification Keywords

H.5.1. Multimedia Information Systems: Animations; H.5.2. User Interfaces: Haptic I/O

INTRODUCTION

Vibrotactile (VT) sensations are becoming more common in user interfaces for a wide variety of applications. In particular, multi-actuator arrays are used in a wide variety of applications and form factors, from chair-based immersive gaming experiences [13] to sleeves for social touch [12] and wearable vests for mobile awareness [16]. Although common, control of these devices is still limited to programming or device-specific tools. Traditionally, designers must be programmers and experts in haptics to creating expressive sensations.

Several tools exist for prototyping or final authoring of haptic sensations. Unfortunately, most authoring tools are designed for a single actuator [6, 21]; those that accommodate multiple

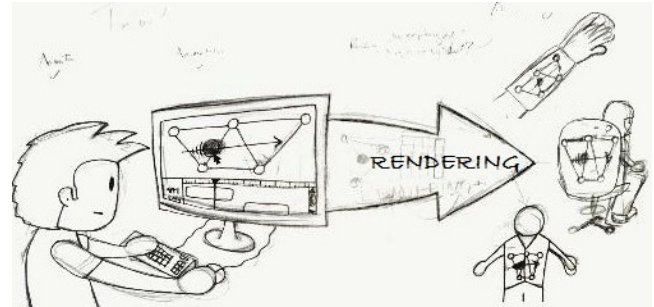


Figure 1: Concept sketch for a haptic animation tool. By creating phantom 2D sensations in both space and time, animators do not need to think in device terms and may be able to design expressive sensations for arbitrary vibrotactile arrays.

actuators force designers into controlling each actuator individually [26, 34].

Conversely, an abstract approach to controlling spatial VT effects would allow designers to think in terms of a single precept rather than multiple device streams, and make VT sensations less device-specific, leading to more proliferation and wider applicability. Specifically, an animation metaphor would allow users with experience making visual designs to smoothly transition into haptic design. Controlling VT sensations in both space and time provides a single mental model that focuses on the sensations, not on the actuators. Furthermore, by playing phantom sensations on an arbitrary VT grid, there is the potential for device independence. We present the concept of a *Haptic Animation Object*, an abstract phantom sensation that can be moved in both time and space, assigned keyframes, and moved anywhere on the grid. See Figure 1 for a concept sketch.

This paper describes the creation of a design tool that features animation objects in a process based on rich input from professional interface designers who already employ haptic feedback in the entertainment industry. We propose an interaction model suitable for large class of haptic displays, and describe its instantiation in *Mango*, a haptic animation tool. We involved users in Mango's design in three steps, the first two informing it and the last evaluating it: (1) *Requirements* gathered from interviews to supplement the literature; (2) *Perceptual Study* comparing rendering algorithms for arbitrary triangulations of VT arrays, the success of which is fundamental to enabling animation objects; (3) *Design Evaluation Study* providing feedback on the overall haptic animation paradigm.

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Our contributions are:

- A set of requirements for haptic authoring tools
- The design and implementation of a haptic animation tool kmEprototype, Mango, including discussions of our:
 - adaptation of an animation metaphor for VT display;
 - conceptual model for device-independent spatio-temporal VT objects, and
 - rendering algorithm and pipeline supporting real-time playback on arbitrary 2-D VT arrays.
- An empirical comparison of interpolation methods for 3-point phantom sensations
- Qualitative feedback from 6 users (4 with animation backgrounds), providing implications for design.

RELATED WORK

Spatial displays of tactors (tactile actuators) are a promising avenue for chair-based immersive gaming experiences [13], sleeves for social touch [12], or wearable vests for mobile awareness [16] and artistic compositions [7]. Here we cover literature on haptic authoring tools and relevant work on vibrotactile communication and perception.

Haptic Authoring Tools

As long as there has been an exploration of haptic or tactile icons [22], there has been a need for compositional tools to facilitate their design [7]. A great deal of previous work has focused on how to prototype, sketch, or control haptic phenomena using non-programming methods.

The hapticon editor [6], haptic icon prototyper [33], posVibEditor [29], and Immersion’s Haptic Studio (www.immersion.com) use a graphical mathematical representation to edit either waveforms or profiles of dynamic parameters (torque, frequency) over time. Features include the combination or multi-tracking of effects, a library of existing effects, and the ability to playback sensations. While these are important control methods, they focus on the low-level control of device features rather than a semantic space, and control devices with either a spatial or temporal component, but not both simultaneously.

The vibrotactile score [18, 20, 21] uses a musical metaphor for controlling vibrotactile sensations. Users can move notes around on a clef, control dynamics through crescendos and decrescendos, and choose different waveforms as different “strings”, allowing the customization of various “instruments” depending on the device and the use case. This was shown to be generally preferable to programming in C and XML [18], but it required familiarity with musical notation.

Recently there has been a trend to add context to the development process. Vivitouch Studio [34] allows for haptic prototyping of different effects alongside video (screen captures from video games) and audio. Unique features include A/B testing and the ability export as a haptic video channel. Immersion’s Touch Effects Studio lets users augment a video with various present tactons on a mobile platform. Although

we focus here on the specific design of haptic sensations, there has also been an effort to automatically extract tactile renderings from video and audio [4, 19, 25].

Several tools have allowed for direct control of phenomena through mobile touch screens. The demonstration-based editor [11] allowed the control of frequency and intensity over a 9-second period. mHIVE, a Haptic Instrument [30] controlled frequency, intensity, waveform, and amplitude envelope of two mirrored output tactors in real time. Both devices were shown to be intuitive and easy to use for exploration or communication, but faltered when refining more elaborate sensations for a set context. Commercially, Apple’s vibration editor has been present in iOS since 2011 (iOS 5).

Additional approaches that aid in the creation of haptic phenomena are haptic sketching [23] for hands-on exploration of haptic ideas in early design, and end-user customization of tactile sensations [31]. In both cases, exploration and broad manipulation is the key, rather than finely controlled end results.

Finally, the control of multi-actuator outputs has been explored by TactiPEd [26] and Cuartielles’ proposed editor [5]. However, these two approaches still embrace the separate control of different actuators, rather than a single semantic sensation produced by the multi-actuator device.

Haptic Icons and Vibrotactile Perception

Haptic or tactile icons, also known as tactons [1, 22], combine brief haptic stimuli with meanings, and offer benefits of privacy and attentional conservation unavailable from their visual and auditory analogs. Main control parameters include frequency, amplitude, duration, location, and to a limited degree, waveform, and they can operate both spatially and temporally. Although these basic parameters have been known for some time, authoring VT sensations has traditionally required the use of programming or the appropriation of audio creation tools for authoring [3, 7]. Research into their design has focused on identifying key low-level control dimensions [7, 10, 22, 35], and on creating memorable, identifiable, and perceptually salient sensations [2, 3] rather than engaging ones.

Several long-known tactile illusions have recently been combined to create a continuous spatiotemporal sensation. The manipulation of vibration duration and stimulus onset asynchrony (SOA) can create the sensation of a continuous tactile sensation known as apparent motion [17]. Meanwhile, manipulating the intensities of two actuators in close proximity creates a single, static output sensation between them, known as a phantom tactile sensation, or funnelling [36]. The tactile brush algorithm [14] combines these illusions to create a single, high-level percept rather than individual actuator activation; this is analogous to specifying a 3-D panning location in a surround sound system rather than editing each audio channel individually. This is a promising approach to create compelling, semantically-based sensations rather than actuator-based vibrations. There are many other illusions that can be exploited in the future to improve these models [8], such as saltation (cutaneous rabbit), kappa tau, and several

	Tools	Notes
E1	After Effects, Maya, Pro Tools	12+ years of experience with haptic perception, engineering, and technologies; uses Max/MSP, OpenFrameworks, Processing, and Visual Studio to create haptic media.
E2	Pro Tools	Professional media designer; went through a six-month long training that includes generation of dynamic haptic experience on seats and supporting platforms using audio and video tools.

Table 2: Interviewed Haptic Experts

others. With many illusions, tactile acuity makes a large difference in the output sensation, and so location and other contexts are critical.

REQUIREMENTS GATHERING

To design Mango, we first gathered two sets of requirements: the first from related literature on haptic authoring tools (“LRs”), and the second from interviews with five industry experts in haptic media creation and animation (“IRs”). Our experts’ profiles are listed in Table 2.

We interviewed two industry experts with haptics experience from a large media company (E1 and E2, Table 2). The objective of these interviews was to determine 1) a conceptual framework for imagining and thinking about haptic feedback, 2) similarities and differences between haptics and other sensory modalities, 3) tools available to generate media content, 4) work process and tools to generate haptic content, and 5) the extent of haptic content.

In addition, we conducted contextual interviews of three industry animators (A1, A2, and A3) using non-haptic animation tools using a think-aloud protocol.

Finally, because we are interested in developing a tool familiar to animators, After Effects, Maya, and Pro Tools formed a second set of design requirements, which often overlapped with the LRs.

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.

Interviewing E1 and E2, and observing A1-3, we independently developed industry requirements (IRs). LRs 2, 3, 4, 5, 6, and 7 emerged independently from these interviews and observations. We report on other requirements focus around media creation tools:

IR1 - Animation window The animation window allows users to draw haptic objects, control them in space, and define its motion path.

IR2 - Timeline During the playback, an animation is played on the window showing the movement of the haptic object. Ob-

ject behaviours are linked to the time track to visualize temporal variations individually. Time tracks are editable by use of inserting key frames, which will allow the user to modify the waveforms on the fly. The window is overlaid with location and type of haptic actuators, providing visual feedback (LR8)

IR3 - Object tools Elaborating on LR2, direct manipulation operations on haptic objects should be analogous to object creation and manipulation (scale, translate) tools in After Effects and Maya.

IR4 - Path tools Be able to define the paths of abstract sensations, e.g., straight lines, curves, traces of an input device, and store these in a library.

IR5 - Haptic Rendering Schemes Once the user plays a pattern, the interface animates the pattern on the animation window and renders tactile feedback on the hardware using a haptic rendering scheme. User selects the rendering scheme from a list defined in the configuration file. The rendering scheme computes output waveforms for each actuator channel.

IR6 - Global parameter tools These tools allow the user to control the overall feel of the haptic pattern, including setting the frequency, intensity, and modulation.

DESIGN

From these two sets of requirements, we developed a design for our tool. In this section, we:

- Describe the interaction with the *animation metaphor*.
- Report the underlying *conceptual & data model*.
- Present our *rendering pipeline* and algorithm.

We based our design primarily on Adobe After Effects and Maya, common visio-spatial controls for special effects and computer graphics (from A1-A3). We also draw from audio-based tools like Pro Tools to control the VT signal (from E1 and E2), and sought breadth in our implementation rather than depth. For example, we chose to implement only linear paths rather than creating a variety of path shapes.

We built Mango with Python 2.7 and Tkinter (Figure 2), which communicated to a haptic pad (Figure 5b) with 10 voice coil actuators via serial communication on USB. We implemented the critical features: animation window and timeline control of both haptic animations and vector sensations, a rendering pipeline providing real-time feedback. Libraries of preset sensations have already been shown to be effective, and it would be premature to include a wide array of effects if the overall paradigm was not finished. Thus, we did not include a library of effects.

Animation Metaphor

In Mango (Figure 2), most interaction is through the animation window (1), and the timeline (2). Our main design theme is an animation metaphor inspired from After Effects and Maya, used by A1-3, combined with the audio (repurposed for haptic) editors (e.g., Pro Tools) used by E1-2. Users are able to manipulate animation objects, abstract phantom sensations that can be moved continuously in space and time, as

Design Requirement	RW	Description
LR1 - Real-Time Playback	[23, 30]	Rapid prototyping is essential for working with VT sensations, especially those without objective metrics of success. By being able to play a sensation or feel it at the time of designing, we can decrease iteration time and develop more engaging sensations. Note, however, that having it too real-time can result in split attention [30].
LR2 - Load, save, manipulate	[15, 28, 30]	A persistent object model is essential for being able to continue to work on sensations for longer projects [30] and for being able to share them 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	[6, 9, 26, 27, 33, 34]	A library of saved sensations is not just an important feature used in previous haptic authoring tools, but is also important for designers in other fields.
LR4 - Device configuration	[18–21, 26]	Because of the many types of VT devices, a general tool must be able to understand different devices. Lightweight configuration files that describe each device are common in the literature. The configuration file allows users to select specific hardware and specify location and type of actuators. Animators can also select a rendering algorithm from a list of available ones.
LR5 - Multiple channels & combination of effects	[6, 26, 29, 33, 34]	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	[5, 26]	Most previous tools consider each actuator as a separate entity. When thinking semantically about a spatial system, a direct view of the layout of actuators and the device is critical for direct manipulation.
LR7 - Audio/visual context	[23, 34]	Haptic perception depends greatly on additional senses TODO: cite . By providing audio and visual feedback that will accompany the feel, these effects can be mitigated and the designer can experience them in context.
LR8 - User Feedback	[30, 34]	Receiving feedback from users, either by demonstration or A/B testing, is extremely valuable.

Table 1: Design concepts for Mango from the literature.

well as *vector sensations*, a way to control each actuator directly and individually, to emulate previous tools. Multiple animation objects and vector sensations can exist simultaneously. The actuator output the sum of all the values generated by the different objects (described later in our rendering algorithm) and sensations.

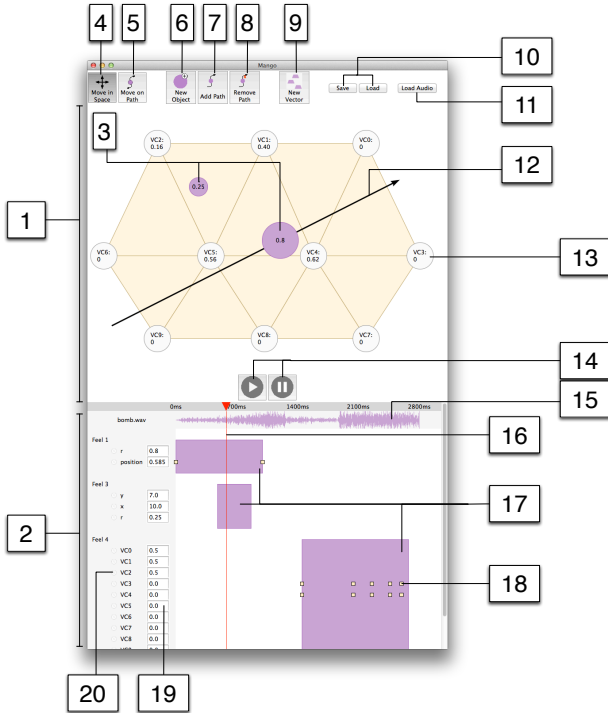


Figure 2: Mango Screenshot **TODO: better labels!**

Animation Window

A user can create a haptic animation object (3) with a dedicated button (6), then manipulate the object in the animation window (1), which shows its position in space for the current time. The output device generates sensations in real time with the rendering algorithm described below, giving the user a sense of what they’ve created.

The animation object has x and y parameters determining position, and an “r” (radius) parameter, corresponding to output voltage (0 to 1), but represented visually with the size of the circle.

An animation object can be given a path (7), or have a path removed (8). An animation object on a path (12) is constrained to be on that path, and instead of having an x and y parameter, they have a single position parameter from 0 to 1. An animation object with a path and can be manipulated in two different ways: moving on path (5), which moves the object from the beginning of the path (position 0), to the end of the path (position 1), and moving in space (4), which moves the path as well as the sensation (keeping the position value the same and mimicking the x/y spatial control of a non-path object). The two modes do not affect the movement of a non-path object, which is always moved in space. The path can be moved by clicking and dragging boxes at each end of the path. Right

now, Mango only supports linear paths, although other forms are easily extrapolated.

Note that, in Mango, each animation object has at most one path, and each path has only 1 animation object (e.g., paths belong to a parent animation object). An alternative model is independent paths, similar to masks in photoshop or After Effects. However, we felt this was too confusing for a first implementation, and the additional power was unnecessary.

Also note that we display visual feedback of the haptic array on the screen. During piloting we found that mirroring the display was critical, i.e., if the user feels a vibration on the right of their back, they should see it on the right of the screen (as if they were looking at the haptic pad from behind).

Timeline

Each animation object is represented in the timeline (2), showing its position in time. The red scrubhead (16) (shown as a triangle and line) shows the current time, and can be dragged around (“scrubbed”). Animation objects can be moved in time by clicking and dragging, and resized to have a shorter or longer duration (17). In addition, individual parameters can be set on the left, by typing values into text fields (19), giving precision. The entire animation can be played and paused using buttons (14), or by pressing the spacebar.

Parameters can be toggled as “keyframeable” or not by clicking on the small clock icon next to their name (20). A keyframeable parameter has a value that depends on the current time. When a value is changed, a keyframe (18) is automatically created at the current time. Values are linearly interpolated between keyframe values; non-linear interpolation (including ease-in and ease-out) is easily extrapolated.

Finally, vector sensations can be created by clicking on a button (9). These sensations control each actuator directly through the parameter values; because each actuator is a parameter, they are also keyframeable. The corresponding actuator is highlighted in the animation window when the text field is selected.

Save and Load

Animations can be saved and loaded (10) to/from JavaScript Object Notation (JSON) files, as they are human-readable and explicit way of storing the workspace. Our data format is described in more detail in the next section. Finally, audio can be loaded (11), and is displayed on the timeline (15). This allows the user to design to a pre-determined context.

Conceptual & Data Model

We have identified three data types (Figure 3): *Haptic animation objects*, high-level device-independent data types for our animation model; *vector formats*, high-level device-specific control common in previous work; and *raster formats*, low-level device-specific formats for rendering and replay.

Haptic animations are high-level specifications of virtual sensations moving in space (in this case, a 2-dimensional plane). High-level parameters, such as location, size, or other semantic qualities, can be constant or vary with keyframes and various interpolation methods. Each haptic animation has

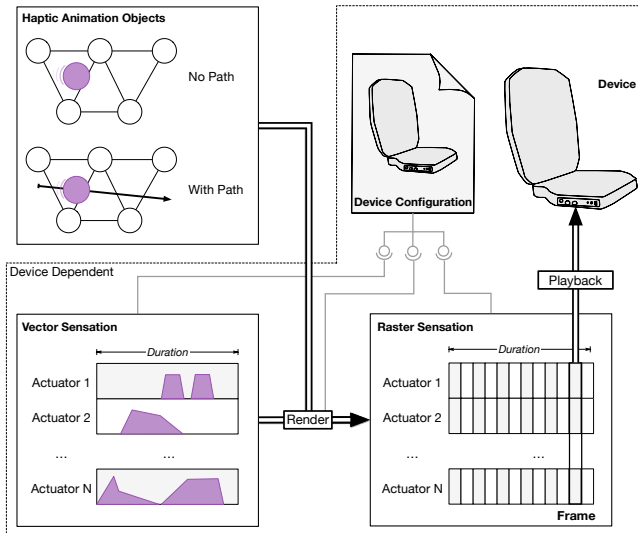


Figure 3: Taxonomy of different vibration descriptions and their representation in Mango.

a start time and a duration. As well, each haptic animation has a type - like visual effects present in animation tools like Adobe After Effects, each animation type is pre-programmed with various parameters. For example, a moving virtual point can have a position, size, and frequency, while a “rain” effect might have a position and more semantic parameters like raindrop frequency or size. Right now, each animation type is programmed as a plugin. In the future, we would like to see end-users be able to define their own parametric control over each part of the device, but this is left for future work.

Haptic animations are device-independent; by modeled each 2-D VT array as a rectangle with width and height, an animation can be played on different devices with just knowledge of its position in that rectangle. However, some animations may need to be adjusted if the aspect ratio changes. Haptic animations can also be combined in novel ways, organized in groups, or generating other haptic animations like a particle generator in a graphical animation tool, and can have paths that constrain motion to a pre-determined trajectory.

Vector formats are similar to those found in previous work (e.g., [6]). Each actuator’s parameters are controlled through activations, with a set of durations, amplitude envelopes (e.g., fade-ins and fade-outs), frequencies, and start times. However, because each actuator is specified, vector formats are device-specific. While vector formats offer finer control than haptic animations (analogous to pixel-level editing of sprites), creating a single percept from independent controls can be challenging. However, multiple vector formats can be multiplexed and overlaid, providing some versatility.

Both haptic animations and vector formats can be rendered into a **raster format**, analogous to a raster-graphics image or WAV file, that is suitable for final playback. A raster format contains a matrix of actuator parameters and frames, the value of those parameters at instance in time. Each raster format also contains a framerate. The playback system finds the

current frame and pushes that to the device, setting each actuator to its desired value. These frames are also used during real-time feedback during authoring; if the playhead remains constant, the same frame will be pushed to the connected device at the desired framerate. In our implementation, all data types are stored as JavaScript Object Notation (JSON) files. Our rendering algorithm is described in the next section.

Rendering Pipeline

Our rendering pipeline first calculates output actuator values for each animation object and vector sensation at that point in time. Because our tool is designed to work with the general class of 2D vibrotactile grids, we wanted to ensure that multiple devices are easily supported, and that feels can be reproduced on different devices. Rendering algorithms describe how to convert a feel into a set of actuator activations for a given device, and how to superpose multiple feels into one final rendering.

Scheme and Device model: To create sensations for arbitrary VT arrays, we first need a general rendering algorithm that automatically handles different configuration files. First, we model each device in a configuration file giving its width and height in physical units (here, centimetres). Then, the format describes each actuator type (e.g., voice coils or rumble motors), giving the control parameters (e.g., frequency, intensity), as well as each parameter’s minimum and maximum allowable value. Finally, the configuration file declares each actuator, giving position (in real world units) and actuator type. See figure TODO for an outline of the configuration file format, here declared as a JavaScript Object Notation (JSON) file.

Vector and raster formats control each actuator individually, Haptic animations, however, must make use of phantom sensations (cite) to display a sensation at an arbitrary spatial location. To do so, we automatically compute a Delaunay triangulation for all actuators on a device. When rendering a sensation at an arbitrary location in-between actuators, we use the barycentric coordinates of the point to determine the relative intensities of each actuator (scaled by the vector sensation’s intensity). See Figure 4 for an example.

Interpolation method: We had to choose an appropriate interpolation method three-point phantom sensations. We began by using barycentric coordinates to represent the desired location in a triangle (Figure 4a). Our approach is to generalize algorithms used in 2-actuator interpolation. There are two main interpolation methods examined in the literature: linear interpolation, better for expressing location but fails to consistently represent intensity, and logarithmic interpolation, which preserves intensity but fails to express a variety of locations [32]. A third option exists, used by the Tactile Brush algorithm [14], which models Pacinian corpuscle sensing by preserving energy as an inverse square law. Figure 4b presents equations for each interpolation method.

In the following study, we empirically compare these possible interpolation methods with the objective of identifying a rendering algorithm that is valid for Mango’s evaluation. This includes the following limitations: curved surfaces may re-

quire additional mapping, as spheres cannot map to planar surfaces without distortion; we use a refresh rate of 100Hz, determined as suitable through piloting, but other rates may be sufficient or higher fidelity; intensity is isotropic for the time being; we only render to a single actuator type, but different actuator types need to be balanced.

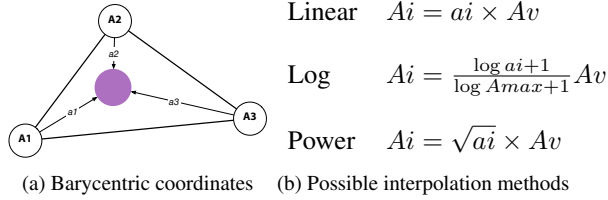


Figure 4: Rendering Algorithm

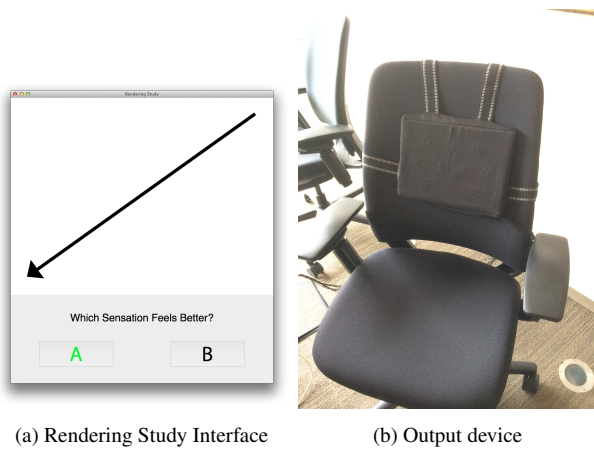


Figure 5: Rendering Study Setup

STUDY 1 - RENDERING ALGORITHM CHOICE

We considered three algorithms for interpolating stimuli between three points: linear, log and power (Figure 4b); then conducted a user study with 18 participants (6 female, ages 21 to 34) to determine which was most effective for rendering in this context. Our focus was on specific optimization rather than a full psychophysical study of 3-actuator interpolation methods.

Methods: Participants were presented with two stimuli delivered on our back display, with identical frequency, intensity, direction, and duration; and made a two-alternative forced-choice between rendering algorithm pairs. The back display (Figure 5a) presented an arrow representing the movement of a phantom sensation in a straight line on the participant’s back. We included two durations (500ms and 1500ms), 8 cardinal directions, and which algorithm was presented first (as stimuli A) and second (stimuli B), balanced, in a randomized order. Frequency was randomly selected from 80Hz, 160Hz, 240Hz, and 300Hz. Intensity was randomly selected between 10 and 20dB, although the 80Hz frequency saturated at 15.3dB (the next highest saturation was 160Hz at about 23dB). 12 extra trials were given as training trials. The output device (Figure 5b) 0dB intensity threshold was calibrated

with two pilot participants for each frequency. A study session took approximately 15 minutes.

Analysis: Analysis was conducted for each pair of algorithm matchups. Because we wanted to control for experimental factors, we fit the data to a logistic regression model with gender, participant id, frequency, intensity, direction, and duration as factors; direction was grouped into three levels: horizontal, vertical, and diagonal. We then performed stepwise regression (specifically, backwards elimination with $\alpha = 0.05$ and a χ^2 to compare between models) to iteratively eliminate factors that were not statistically significant, then analyzed the resulting model.

Log vs Linear We eliminated 5 factors: duration ($p = 0.970$), frequency ($p = 0.834$), intensity ($p = 0.133$), and direction ($p = 0.106$). Participant id was significant ($p = 0.000314$), suggesting individual differences in algorithm choice. Gender was removed as it is subsumed by participant id. The resulting model had a goodness-of-fit of ($R^2 = ???$). 95% confidence intervals for each participant were computed using the model, using Bonferroni correction to control for multiple comparisons. Of the 18 participants, 11 (participants 1-5, 7-9, 12-14) were significantly more likely to pick log over linear; none were more likely to choose linear than log.

Log vs Power We eliminated all 6 factors: intensity ($p = 0.949$), participant id ($p = 0.875$), frequency ($p = 0.940$), direction ($p = 0.725$), duration ($p = 0.233$), and gender ($p = 0.119$). As such, we computed the overall 95% confidence interval of the proportion of people selecting log over power as 37.06% to 87.40%, suggesting no significant difference of preference between log and power.

Power vs linear We eliminated 4 factors: intensity ($p = 0.865$), direction ($p = 0.587$), duration ($p = 0.384$), and gender (subsumed by participant id); the final fitted model had a goodness-of-fit of $R^2 = ???$. Thus, preference for power compared to linear depended on individual differences (participant ids) and frequency, with higher frequencies increasing the likelihood of preferring power to linear (the coefficient is positive). We computed confidence intervals for each participant-frequency combination using the fitted model, controlling for multiple comparisons using Bonferroni correction. Only 22 out of 72 participant-frequency combinations selected power over linear more than 50% of the time; none chose linear over square root more of the time.

Results: Of the three algorithms, we chose to use logarithmic interpolation, as 11 out of 18 participants reported that it felt more like the intended motion than linear, and no significant difference was found between log and power. Further, power and linear display a more complicated relationship involving frequency that we leave to future work to investigate.

STUDY 2 - PARADIGM EVALUATION

In our second study, we implemented main functional features of Mango to investigate the animation paradigm and compare it to actuator-level control.

Protocol

We conducted a qualitative study using our Mango prototype (Figure 2). 6 participants (P1-6, 3 female) were introduced to the device.

First, each participant was interviewed about their background. P2, 3, 4, and 5 had animation experience but little experience with haptics; P1 had experience with haptics but not animation beyond video editing; P6 had no experience with haptics or animation, but was familiar with computer media tools like Photoshop.

Next, each participant given a demo of how to use haptic animation objects and vector sensations, and instructed to design an alert for each one. Half the participants were introduced to the animation objects and conducted that alert first; the others were introduced to vector sensations first.

Then, each participant was given three design tasks:

1. **Spatial** - Tell a driver to turn left.
2. **Temporal** - Create a heart beat.
3. **Context-based** - Create a sensation to match a sound. In this case, a 3 second sound effect of a bomb falling (with a whistle descending in pitch) then exploding. This sound was chosen because it was complex (with two main components), and because the first part might be interpreted as spatial (as pitch is analogous to location [7]), while the second is more temporal.

After, a post-study interview was conducted with the participant to understand their subjective experience working with the tool. They were asked to explain their experience in as full detail as possible, as advised by phenomenological protocols [24]. The interviewer then followed up on interesting statements or concerns. Later in the interview, the interviewer asked the participant to compare animation objects with vector sensations, and to walk through the interface to get more feedback.

Results

Results were analyzed by considering each statement, memoing, coding, and relating codes according to Corbin and Strauss's method.

Theme 1 - Animation Metaphor

All 6 participants were able to accomplish all five design tasks (object alert, vector alert, heartbeat, left turn, sound) using Mango. All six provided positive feedback about the interface, describing it as intuitive (P1-5), and agreeing that it was an animation tool: *"Its up to the standards of other animation tools"* (P1), *"This is totally animation"* (P2), *"It felt very much like an animation tool"* (P4), *"Im 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 around not implementing enough of the features of their preferred tools and other negative transfer, as well as other general polish problems: *"gotta spline [the keyframe interpolation]"* (P2), *"a couple quirks butthere was nothing difficult to overcome"* (P4), *"being able to design your own curve [path] would be really nice"* (P5).

Theme 2 - Animation objects preferred to Vector Sensations

Participants used animation objects more frequently than vector sensations: P5 was the only one to choose a vector sensation (in his sound task); he combined it with an animation object. P1 started with a vector sensation with the heartbeat, then switched to animation object for her remaining tasks; no other participants used vector.

Animation objects were described as easier to use and more intuitive: *"After using the new object Id probably never use new vector again"* (P2), especially to describe motion or position: *"easier to find the location of the heart"* (P1). They were also described as more appropriate for people without animation experience: *"If I werent an animator I think I would only use [animation objects]"* (P4), *"You have to be a little more careful when animating [vector sensations]"* (P5).

Animation objects and vector sensations both supported different workflows. Animation objects tended to be described as better for position, movement, and if you wanted to have multiple objects, while vectors were preferred for more fine-tuned control, as easier when position and movement didnt matter as much, and supported a large technique of fewer objects with many keyframes. *"You can control multiple [actuators] at the same time, so you dont have to create new objects and then put them everywhere on the screen"* (P1), *"[Animation objects] can be more comfortable to use when one doesnt work with keyframes"* (P3), *"If you want precise control over [actuators], then vector is the way to go"* (P4).

Theme 3 - Replication in Space and Time through Copy and Paste

Replication in both space and time was common while interacting with Mango. Many designs had symmetrical paths to reinforce sensations; see P2s sound design (Figure 6a) for an example exemplifying space. 4 participants (P1-3,5) requested copy and paste as a feature, suggesting it would be useful (P2, P3), faster (P1, P2), and easier (P5). P1, P2, and P5 all wanted to duplicate their heartbeat sensation to be multiple beats, but did not do so without copy and paste, instead saying they would repeat or loop it.

Theme 4 Feedback, Context and Imitation

Participants found having visual or audio feedback valuable, and responded positively to the sound task: *"I was really happy with the bomb one, because I could really heart 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). More generally, additional context was valuable to participants. During the heartbeat task, P1 stopped to think about her own heartbeat and P2 used a YouTube video of a heartbeat as a reference point P3 especially said her design was guided by imitating things in for the non-sound tasks; *"Its typical to have two beeps for mobile phones"* (P3).

Similarly, the real-time haptic feedback was valuable to participants. Participants would drag the animation objects

around both in space (in the animation window) and by scrubbing at various speeds 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). P2 even noted that YouTube did not have the same scrubbing feedback as Mango: “*I wish I could scrub back and for right here*” (P2).

Of course, while adding haptic context is important, constant vibrations were annoying. Mute for haptic feedback might be a welcome feature. P4 and P5 were observed to move the timeline so that no output was playing while they were entering values. “*It would be nice if when I [enter values into text fields] it doesnt go off constantly, its getting annoying*” (P3). It is unclear whether muting should be allowed for each object (much like hiding layers in photoshop), or if a global mute is more important; of course, theres no reason not to have both.

Examples of Animations

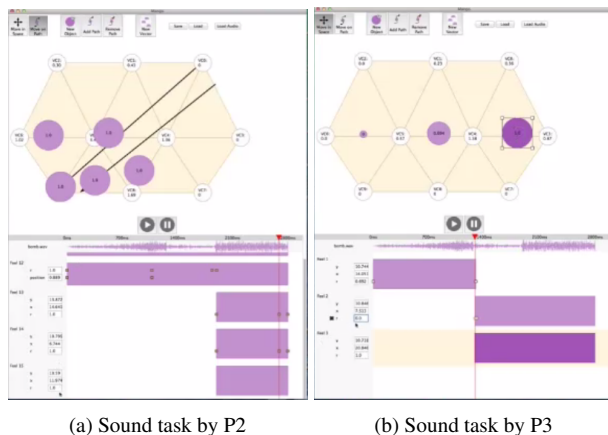


Figure 6: Examples of Animations

IMPLICATIONS FOR DESIGN

Overall, haptic animation represents a promising new direction to support haptic design.

Haptic Animation is a promising metaphor. Described as easy and intuitive, haptic animation objects not only provide an expressive way for animators to transfer their knowledge, but a possible avenue for cross-device haptic authoring.

Single vector sensation control. Participants found vector sensations, or actuator-level control, less intuitive but more powerful. However, participants rarely created multiple vector sensations and attempted to click on the actuator visualizations in the animation window. We suggest that instead of having multiple vector sensations available to the animator, a single, omnipresent vector sensation could be manipulated both from the timeline and the animation window.

Realtime feedback is valuable, but haptics output needs a mute feature. Audio-visual context, such as that in [34], was shown to be valuable. Real-time feedback for haptics also facilitated exploration, but became annoying when left on too long.

Support repetition with copy and paste. Although libraries of presets were known to be valuable, we found that copy and paste is essential to support repetition (both in time and space) in haptic animation.

Triangulation-based rendering can represent position and motion. Although a full psychophysical evaluation is out of this paper’s scope, we found that we could represent motion and position continuously on a 2D VT grid using our rendering algorithm. As well, logarithmic interpolation tentatively seems like the most preferred method.

CONCLUSION

In this paper, we surveyed and synthesized design requirements from the literature and industry designers, proposed an interaction model suitable for large class of haptic devices (2D VT arrays), defined the design space for authoring tools of VT arrays, presented a validated rendering algorithm for arbitrary triangulations of VT arrays, designed a working prototype for general VT authoring based on animation authoring tools, and evaluated our design with a user study.

Future work will focus on the other parts of the design space, including viewpoints for user feedback and more in-depth analysis of user needs.

ACKNOWLEDGMENTS

Withheld for anonymous submission

REFERENCES

1. Brewster, S., and Brown, L. M. Tactons: structured tactile messages for non-visual information display. In *5th Australasian User Interface Conference (AUIC '04)* (Jan. 2004), 15–23.
2. Chan, A., MacLean, K., and McGrenere, J. Learning and Identifying Haptic Icons under Workload. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE (2005), 432–439.
3. Chan, A., MacLean, K., and McGrenere, J. Designing haptic icons to support collaborative turn-taking. *International Journal of Human-Computer Studies* 66, 6 (2008), 333–355.
4. Chang, A., and O’Sullivan, C. Audio-haptic feedback in mobile phones. In *CHI '05 extended abstracts* (Apr. 2005), 1264–1267.
5. Cuartielles, D., Goransson, A., Olsson, T., and Stenslie, S. Developing Visual Editors for High-Resolution Haptic Patterns. In *Workshop on Haptic and Audio Interface Design (HAID) Posters and Demos* (2012), 42–44.
6. Enriquez, M., and MacLean, K. The haptic editor: a tool in support of haptic communication research. In *Haptics Symposium (HAPTICS '03)*, IEEE Comput. Soc (2003), 356–362.
7. Gunther, E., Davenport, G., and O’Modhrain, S. Cutaneous grooves: composing for the sense of touch. In *NIME '02* (New York, NY, USA, May 2002), 1–6.

8. Hayward, V. A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store. *Brain research bulletin* 75, 6 (Apr. 2008), 742–52.
9. Herring, S. R., Chang, C.-C., Krantzler, J., and Bailey, B. P. Getting inspired! Understanding How and Why Examples are Used in Creative Design Practice. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, ACM Press (New York, New York, USA, Apr. 2009), 87–96.
10. Hollins, M., Faldowski, R., Rao, S., and Young, F. Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis. *Perception & Psychophysics* 54, 6 (Nov. 1993), 697–705.
11. Hong, K., Lee, J., and Choi, S. Demonstration-based vibrotactile pattern authoring. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*, ACM Press (New York, New York, USA, Feb. 2013), 219.
12. Huisman, G., Darriba Frederiks, A., Van Dijk, B., Hevlen, D., and Krose, B. The TaSSt: Tactile sleeve for social touch. In *2013 World Haptics Conference (WHC)*, IEEE (Apr. 2013), 211–216.
13. Israr, A., Kim, S.-C., Stec, J., and Poupyrev, I. Surround haptics: tactile feedback for immersive gaming experiences. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12*, ACM Press (New York, New York, USA, May 2012), 1087–1090.
14. Israr, A., and Poupyrev, I. Tactile brush: drawing on skin with a tactile grid display. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, ACM Press (Vancouver, BC, May 2011), 2019–2028.
15. Johnson, J., and Henderson, A. Conceptual models: begin by designing what to design. *interactions* 9, 1 (Jan. 2002), 25–32.
16. Jones, L., Nakamura, M., and Lockyer, B. Development of a tactile vest. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.*, IEEE (2004), 82–89.
17. Kirman, J. H. Tactile apparent movement: The effects of interstimulus onset interval and stimulus duration. *Perception & Psychophysics* 15, 1 (Jan. 1974), 1–6.
18. Lee, J., and Choi, S. Evaluation of vibrotactile pattern design using vibrotactile score. In *Haptics Symposium (HAPTICS '12)*, IEEE (Mar. 2012), 231–238.
19. Lee, J., and Choi, S. Real-time perception-level translation from audio signals to vibrotactile effects. In *CHI '13* (Paris, France, Apr. 2013), 2567–2576.
20. Lee, J., Ryu, J., and Choi, S. Graphical authoring tools for vibrotactile patterns. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE (2009), 388–389.
21. Lee, J., Ryu, J., and Choi, S. Vibrotactile score: A score metaphor for designing vibrotactile patterns. In *Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (World Haptics '09)*, IEEE (Mar. 2009), 302–307.
22. Maclean, K., and Enriquez, M. Perceptual design of haptic icons. In *In Proceedings of Eurohaptics* (2003), 351–363.
23. Moussette, C., and Banks, R. Designing through making. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11)*, ACM Press (New York, USA, Jan. 2011), 279–282.
24. Moustakas, C. *Phenomenological Research Methods*. SAGE Publications, Inc., 1994.
25. Myongchan Kim, Lee, S., and Seungmoon Choi. Saliency-Driven Tactile Effect Authoring for Real-Time Visuotactile Feedback. *LNCS: Haptics: Perception, Devices, Mobility, and Communication 7282* (2012), 258–269.
26. Paneels, S. A., Anastassova, M., and Brunet, L. TactiPEd: Easy Prototyping of Tactile Patterns. *INTERACT 2013 8118* (2013), 228–245.
27. Paneels, S. A., Roberts, J. C., and Rodgers, P. J. HITPROTO: a tool for the rapid prototyping of haptic interactions for haptic data visualization. In *2010 IEEE Haptics Symposium*, IEEE (Mar. 2010), 261–268.
28. Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., and Eisenberg, M. Design principles for tools to support creative thinking. In *NSF Workshop Report on Creativity Support Tools*. (Washington, DC, 2008).
29. Ryu, J., and Choi, S. posVibEditor: Graphical authoring tool of vibrotactile patterns. In *2008 IEEE International Workshop on Haptic Audio visual Environments and Games*, IEEE (Oct. 2008), 120–125.
30. Schneider, O. S., and MacLean, K. E. Improvising Design with a Haptic Instrument. In *Haptics Symposium (HAPTICS '14)* (Houston, USA, 2014).
31. Seifi, H., Anthonypillai, C., and MacLean, K. E. End-user customization of affective tactile messages: A qualitative examination of tool parameters. In *Haptics Symposium (HAPTICS '14)*, IEEE (Feb. 2014), 251–256.
32. Seo, J., and Choi, S. Initial study for creating linearly moving vibrotactile sensation on mobile device. In *2010 IEEE Haptics Symposium*, IEEE (Mar. 2010), 67–70.

33. Swindells, C., Maksakov, E., MacLean, K., and Chung, V. The Role of Prototyping Tools for Haptic Behavior Design. In *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '06)*, IEEE (2006), 161–168.
34. Swindells, C., Pietarinen, S., and Viitanen, A. Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. In *Haptics Symposium (HAPTICS '14)*, IEEE (Feb. 2014), 515–521.
35. Ternes, D., and MacLean, K. E. Designing Large Sets of Haptic Icons with Rhythm. *LNCS 5024 - Haptics: Perception, Devices and Scenarios 5024* (2008), 199–208.
36. v. Bekesy, G. Sensations on the Skin Similar to Directional Hearing, Beats, and Harmonics of the Ear. *The Journal of the Acoustical Society of America* 29, 4 (June 1957), 489.