Towards a framework for ubiquitous audio-tactile design

Maximilian Weber, Charalampos Saitis

Towards a framework for ubiquitous audio-tactile design

Maximilian Weber[1] and Charalampos Saitis[2]

[1]Audio Communication Group, TU Berlin, Berlin, Germany
[2]Centre for Digital Music, Queen Mary University of London, London, UK

To enable a transition towards rich vibrotactile feedback in applications and media content, a complete end-to-end system — from the design of the tactile experience all the way to the tactile stimulus reproduction — needs to be considered. Currently, most applications are at best limited to dull vibration patterns due to limited hard- and software implementations, while the design of ubiquitous platform-agnostic tactile stimuli remains challenging due to a lack of standardized protocols and tools for tactile design, storage, transport, and reproduction. This work proposes a conceptual framework, utilizing audio assets as a starting point for the design of vibrotactile stimuli, including ideas for a parametric tactile data model, and outlines challenges for a platform-agnostic stimuli reproduction. Finally, the benefits and shortcomings of a commercial and wide-spread vibrotactile API are investigated as an example for the current state of a complete end-to-end framework.

Introduction

As vibrotactile feedback becomes more widespread in telecommunication, media, software, and general application design it is desirable to define methods and principles that allow for a comfortable way to design, integrate and transport tactile media assets, without having to worry about any platform or device specific constraints. It is also important to define metrics for quality and testing standards before euphemistic marketing terms, such as “HD-Haptics” lose their meaning, as there is no definition or agreement on what this term encompasses. A similar challenge in defining metrics could be observed in the 1950s, giving rise to the high fidelity (HiFi) audio era [6].

Initial steps towards standardizing and evaluating haptics (including the vibrotaction) have been made in parts of ISO 9241 [17, 16, 18]. While these parts of the ISO standard provide high-level guidelines for the integration of haptic modalities (i.e., tactile and kinaesthetic [39]), a concept for a full-stack solution achieving these standards is yet to be formalized and evaluated. Furthermore, the ISO standard at it’s root mainly portraits an ergonomic perspective on haptics with a focus on how to design user-initiated interactive task primitives and interaction elements [17]. While care has been given to ensure perceptual, information encoding, and systematic parameters are addressed [16] there is a lack of detail on how the information inherent to a tactile stimulus is intended to be composed and how this information is accurately reproduced across various tactile display technologies. This issue is especially important during a transitional period, in which the integration of adequate high performance hardware is slowly adapting and most devices still run on legacy eccentric rotating mass (ERM) actuators or linear resonant actuators (LRA), that mostly appear to be designed for power efficiency instead of the quality of experience.

The sender-to-receiver signal fidelity is an important factor to ensure that a stimulus for a designated vibrotactile event is displayed the way originally intended by the designer of an experience. For a closed system application this is achievable since every component in the transmission and reproduction path can be controlled by the designer. As soon as we want to establish a platform for third parties to create tactile content, while also ensuring cross-platform support across devices, this becomes a complex task: Not only is there a lack of standards for storing and transmitting tactile stimuli, it is also not guaranteed that a stimulus is played back correctly due to a variety of tactile application programming interfaces (APIs), interaction-specific interference and various tactile stimulation technologies on the market.

Standardizing a vibrotactile framework could benefit many use cases and accelerate the development of applications in the entertainment, utility and medical sectors. Various research branches could benefit from a standardized process by improving the exchange of tactile stimuli assets, measurement data, testing procedures and parameters. Especially the possibility of remote empirical research, similar to the use of browser-based listening tests, could be facilitated by enabling a standardized vibrotactile framework for end user devices.

This work first discusses challenges in (audio-)tactile stimuli design and reproduction, then proposes a theoretical framework addressing these issues. Furthermore, the CoreHaptics API by Apple is used as an example for a widely available vibrotactile framework and is discussed with regards to the proposed framework.

Background

Natural audio-tactile events

To form a coherent percept of the environment, an object or event, our brain combines information from various senses [37]. For an auditory-tactile experience the integration of both modalities occurs early and close to primary sensory areas, as experiments using functional magnetic resonance imaging (fMRI) scans of primate brains have shown [21]. Joint audio-tactile percepts are a common and a natural phenomenon. If we consider an acoustic event, it is possible to not only sense the acoustic waves propagating from the event with our ears, but also experience a skin deformation if enough energy is present. Sound waves can also propagate through structures, such as the ground or other objects we are in contact with and thus lead to a tactile stimulus. Events experienced this way are completely *passive*, meaning that no active participation or action is required by a subject to experience them. A common example is the vibration experienced at a concert [27] or while driving a car. For interactive events, such as probing an object’s geometry, contours, and texture, we integrate kinaesthetic, tactile and proprioceptic information into what is considered to be an *active* event. Here, forming a vibrotactile percept requires active participation by the subject while the velocity and direction of movement have an influence on the characteristics of the vibrotactile stimulus, thus integrating both the proprioceptic information on velocity and the vibrotactile stimulus. For both passive and (inter-)active tactile events it has been shown that an integration of both auditory and tactile information play a significant role in forming a percept by displaying various effects on the cognition or perceived quality of an event [26, 35, 34, 19, 40, 38, 12, 28, 14, 2].

When designing a vibrotactile stimulus it is important to consider the consequences for both *passive* and *active* feedback and how strongly the virtual or mediated interaction mirrors a *natural* event. Pressing a virtual button, for example, requires only a short (active) interaction which enables a single event call to trigger a corresponding tactile stimulus simulating the natural tactile response of a button. On the other hand drawing in a virtual paint application, for example, requires a continuous event by simulating the interaction between pen and surface that is virtually drawn on. Such an interaction could not merely be triggered by a sole binary event but would require a continuous synthesis of the desired tactile event, until contact with the surface is broken. In both cases though, using sound as a source for vibrotactile stimuli design seems feasible for many virtual interactions, as the resulting vibrations can conceptually not only be sensed by the mechanoreceptors, but also propagate to the ear to form an integrated percept of the interaction (and vice versa).

It thus becomes apparent that not only the design of the stimuli itself is important, but also the plausibility of the logic that triggers it on a software level (cause and effect). The same procedure is already conducted when designing sound for a virtual environment or application and some applications are designed especially for this workflow (f.e., FMOD, Wwise). An environment to effectively design audio-tactile or multimodal interactions according to this workflow, while enabling the composition of a corresponding vibrotactile stimulus has yet to be developed. Integrating tactile assets alongside the already existing sound design workflow and deriving tactile assets from the already existing audio assets therefore seems like a feasible option.

The strong integrated connection between auditory and tactile stimuli is a concept getting increased attention in the design of musical interfaces, and has been urged to be improved due to the "veil of tactile paralysis" between the musician and the sound source [24, 33, 3]. While digital instruments have been reported to be "lifeless" and "cold" compared to their analog counterparts, the addition of an artificial bodily resonance to simulate natural tactile events has indicated to be a remedy for the missing "warmth" in digital musical interfaces. This could have the potential to improve the tightly interlocked feedback loop between virtuous musicians and their (digital) instrument — especially when the musician is in a loud environment and can't rely on the auditory response of the instrument alone. Recently, digital synthesizers, such as the OP-Z by Teenage Engineering, have been countering the lack of bodily resonance by giving life to their product using a so-called "Rumble module"[1]. The "veil of tactile paralysis" is arguably present in many other applications beyond musical interfaces, such as video games and movies, which could equally benefit from solutions using audio assets as a starting point.

We can conclude, that the workflow for designing and integrating audio assets for applications and media is comparably well understood and widespread throughout the industry. This workflow could serve as a blueprint for the design and integration of vibrotactile assets, while providing a starting point for the design of the vibrotactile assets themselves. The framework proposed in this work therefore aims to utilize existing audio assets as a starting point for tactile stimulus design and integrate tactile assets alongside the already existing logic from the sound design process. Using audio assets, that are already present in an application, removes the need to start designing tactile assets "from scratch" and automatically keeps all modalities in perceptual synchronicity, ensuring that the cross-modal integration of information found in natural audio-tactile events is warranted.

[1] TE's "Rumble Module": tactile feedback for the OP-Z synthesizer
`https://teenage.engineering/products/op-z/modules/rumble`

Previous work on audio-tactile translation

Translating audio to tactile stimuli has been the subject of previous works researching mostly the joint (i.e. bimodal) display of audio-tactile stimuli [31, 27, 3, 32, 9]. Within these works various methods for audio-tactile signal translation are explored. As the perceptual frequency ranges of auditory and vibrotactile stimuli overlap, the most straightforward method would merely requires enough energy to be present in the tactile sensitivity range from 30 to 1000 Hz in a PCM encoded signal, such as the one contained in a WAVE file. This signal can then be downsampled and low-pass filtered at approximately 1 kHz to be played back by the actuator. An issue with this method is, that the tactile display on the receiving end might not be capable of reproducing such a, comparably wide-band, signal with enough fidelity to match the intent of the designer. Furthermore, a tactile stimulus signal obtained this way includes a lot of irrelevant and redundant information due to the comparably low resolution of the tactile sense [13, 10, 11, 29] and could thus be compressed, if the transmission bandwidth or data storage is a concern. If the audio source lacks meaningful content in the tactile perceptual range, there are various ways to augment the signal. One option is to pitch-shift the signal downwards until a desired effect is achieved [31, 20]. This method works well if the content of the pitched-down signal is representative for the rest of the signal content and reflects the intended experience. Otherwise further filtering, editing and aesthetic augmentation is required. Using knowledge and tools from the sound design process could aid in making these aesthetic decisions before further encoding the tactile stimulus from the audio source.

To augment the low frequency range of an audio source and to achieve a higher level of parametric control over the temporal trajectory of the stimulus, a combination of an envelope follower and a signal generator can be used [27]. Here both the parameters of the envelope follower and the pitch of the signal generator can be controlled independently until a satisfying result is achieved. Depending on the audio source material and the fidelity of the reproduction system (i.e. tactile actuator) the signal synthesized this way can then be combined with the original audio source by adding the synthesized perceptual frequency content if needed, while not compromising on more complex timbral information from the source signal. The tactile signal representation using an envelope follower is inherently monophonic and therefore can't model the entire perceptual range sufficiently. It is feasible to track the temporal energy trajectory of a signal this way but it neglects (complex) changes in the frequency domain over time.

For an experiment in speech recognition, a vocoder approach utilizing 16 solenoid actuators was utilized. The 16 channel filter bank of the vocoder ranged from 200 to 8000 Hz in third octave spacing. Each solenoid was driven by a 100 Hz square wave modulated by the energy of each filter channel [7]. Through this method an abstract imprint of the audio (here: speech) signal's spectrum was created while a actual wideband reproduction was not utilized. This might have not been feasible due to technological limitations in actuator technology at the time — which today is still a comparably rare technology to find on the market.

By utilizing the four channel theory of touch [4], a translation method catering towards the four individual types of mechanoreceptors has been proposed [3]. Here, a set of audio analysis features are mapped to a set of dynamic synthesis parameters: the spectral centroid [22] of the audio spectrum was mapped to the pitch of a signal generator in a perceivable range from 40 to 400 Hz. The spectral flatness of the signal was mapped to an equal power cross fade between a sine wave and a square wave — more tonality in the audio source was therefore represented with a richer harmonic spectrum in the vibrotactile domain. Finally the amplitude of the tactile signal is modulated by a envelope generator with a adjustable decay. This method illustrates a way to reflect timbral changes in the tactile range by utilizing information on the spectral envelope of the audio source while still allowing for a design choices by adjusting the analysis and synthesis parameters. This method doesn't capture the exact momentary spectral content by reflecting spectral change in the audio source merely by the flatness of the spectral envelope and approximating the flatness in the tactile domain by inducing harmonic content in form of a square wave. While this method was intended to add vibrotactile feedback to an electroacoustic instrument it might prove to be insufficient for a standardized approach in encoding, storing, transmitting and decoding a tactile signal due to the shortcomings in the spectral shape.

A commonality of the audio-tactile translation methods reviewed here, is that they make use of the comparably low resolution of the tactile sense by abstracting (or decomposing) the audio signal into a parametric representation. Interestingly, popular APIs and SDKs for game and application development utilize similar parametric representations to define vibrotactile patterns. This indicates, that achieving an adequate parametric representation of a tactile stimuli from an audio source (i.e., *tactile data model*, see Figure 3) can be useful and potentially be matched with a standardized API, which would provide a platform-agnostic interface to drive various vibrotactile display technologies.

Challenges in tactile reproduction

Assuming that a standardized data model and API exists leaves further challenges in the reproduction of a stimulus due to physiological and hardware specific characteristics. The following sections provide a theoretical model and observations that aim to guide the decision making during the design of vibrotactile devices. Furthermore, standardized measurement methods are proposed to characterize device specific parameters that are required to ensure that a vibrotactile device can be used to it's best potential by a standardized API.

Bodily interference on vibrotactile systems

A device incorporating vibrotactile feedback can afford a range of user interactions [30]: It can either be attached to the human body in form of a wearable device or afford a variety of grasp interactions, as commonly found in smartphones or game controllers. Attaching a device or performing a grasp interaction can dynamically change the system properties by inducing bodily interference on the vibrotactile display system. Previous works have identified this issue for a sitting interaction (i.e. a stool or chair form factor) and proposed a Body Related Transfer Function (BRTF) to analyse and counteract bodily interference [1]. While deriving a transfer function for the entire body is a good approach for some applications, devices that afford other forms of bodily contact might require a similar approach for the individual application. This motivates the use of standardized measurements and filter design methods (i.e. motor control) to compensate potential bodily interference on a vibrotactile reproduction system [25, 8]. The model described in this section is thought to be used as a theoretical framework, that aims to help understand where bodily interference comes from and why it is important to ensure a high fidelity stimulus reproduction. It is thought to provide a tool to aid discussions around interaction-, product- and tactile experience design.

The force needed to create a vibrotactile stimuli in a device can be modeled using an approximate linear spring-mass system for most actuator models. A spring-mass system can induce a force vector $\vec{F_0}$ by accelerating a moving mass m_0 following Newton's second law of motion ($\vec{F_0} = m_0\vec{a}$). The method for creating the acceleration $\vec{a}$ is dependant on the actuator specific design. The force of the actuator is opposed by both the application device $\vec{F}_{app}$ and the skin tissue $\vec{F}_{skin}$. If a rigid connection between the actuator and the device is ensured the stiffness k_{app} and dampening d_{app} coefficients vanish — leaving only the device mass m_{app} to be considered as the vibrations propagates through the device to the human skin. By exerting the resulting force from the device to the skin, i.e. setting the skin tissue into motion and deforming it

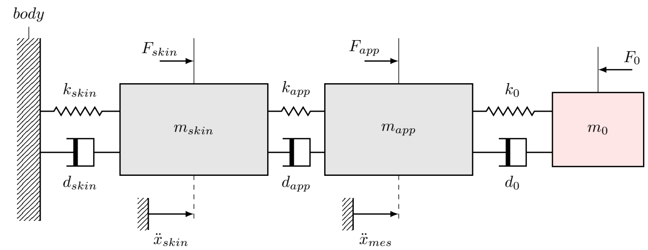


Figure 1: The figure shows a linear mechanical model of the physical interaction between a vibrotactile display devices and the human physiology (i.e. skin tissue).

(note the acceleration vector $\ddot{x}_{skin}$ in Figure 1) the mechanoreceptors are excited and allow the somatosensory cortex and higher level brain regions to form a percept that can be experienced. The region of the skin that is set into motion can be modeled by a mass m_{skin}, an elastic stiffness k_{skin} and viscous dampening d_{skin} that counteracts the force of the vibrating device. Note that the theoretical linear model presented here is a simplification, as the interaction with the skin usually shows non-linear behaviour. The acceleration vector $\ddot{x}_{mes}$ indicates the point in the system, where actuators within a device are commonly measured.

To conclude, the theoretical model presented here suggests that linearizing an actuator based on it's isolated characteristics might not be enough to ensure an accurate reproduction of a tactile stimulus when embedded in an end user device. When designing a vibrotactile device or application, the mass and form factor of the tactile display device, as well as the interaction with the human skin can alter the actuators response and should be considered. Practically this suggests that efforts towards linearization would ideally be conducted on a device level or in the final configuration, depending on the form of interaction. To ensure a platform-agnostic tactile data model, this observation further suggests, that the characteristics of the tactile device need to be provided on the device for an accurate resynthesis of the tactile stimulus. In a forthcoming publication we discuss the efficiency of inverse filtering for varying bodily-induced interference by an exemplary use case of a tactile wristband on various test subjects with varying physiology and comfort levels (tightness and location of the wristband strap). It could be argued that these measures are an important aspect to make an application inclusive and resistant to gender or physiology induced biases. A technology should ideally be designed around all possible users and minimize the amount of negative experiences for individuals due to technological shortcomings, which could be induced by a lack of physiological variety in user tests (cf. [5]).

Variability in vibrotactile actuators

To ensure a high fidelity reproduction of a stimulus it is not only important to save and transmit all relevant information of the intended tactile design to the reproduction system, but also the capability of the reproduction system to play back the tactile signal as precisely as possible. Next to the bodily-induced interference described above, the variety of actuator models on the market indicate strong differences in both their individual resonant frequencies and the resulting bandwidth and thus influence the characteristics of the resulting stimuli. The linear resonant actuators (LRA) found in modern devices are mostly highly underdamped. This characteristic results in a narrow resonant peak and therefore only a narrow band playback — which in turn is only partially useful for wide-band tactile stimuli reproduction (see Figure 2).

To illustrate the inherent variance between current smartphone devices, due to the built-in actuator technology, a set of impulse response measurements were conducted. First, all smartphones listed in Figure 2

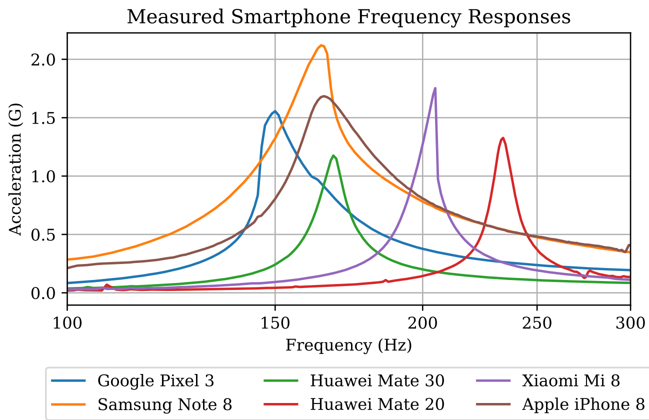


Figure 2: The figure shows plots of acceleration responses from current smartphone models on the market. The resonance characteristics, the variance in the resonant frequencies of the devices (i.e. actuators) and the inability to generate a reasonable acceleration within the entire tactile perceptual range from 30 to 1000 Hz for these devices becomes apparent.

were carefully modified to make the contained actuator's connectors accessible from the outside. Next the driving voltage for each actuator was calibrated according to their nominal voltage. This was done by researching the data sheet from the actuator's manufacturer or manual-heuristically by stepping up the voltage until a measurable distortion was present in the resulting frequency spectrum. Unfortunately there is no standardized value for an acceptable amount of total harmonic distortion (THD%) in tactile devices so far. Some devices showed distortion levels up to 25% THD even when driving them at nominal voltage. For reference standardized loudspeaker measurements have acceptable distortion levels of 1%, 3% or 10% [6]. The measurements illustrated here are merely meant as an exemplary representation of the variability in the market today, which exceeds the smartphone market and is seen in various other applications, too.

Even though the human tactile sensitivity ranges from 30 to 1000 Hz, the devices measured in Figure 2 all cover a comparably narrow frequency band and dynamic range due to their strong resonant peaks in the range from 150 to 240 Hz. These characteristics indicate a poor performance in reproducing wide band signals and therefore limit the quality of reproduction (i.e. fidelity) of a tactile stimulus. Due to the strong resonant characteristics of the actuators within the devices it is merely feasible to recreate a single frequency mapped to the resonant frequency of the device due to the lack of a useful dynamic range outside of resonant peak. This design is most likely done on purpose, as its most efficient to drive an actuator at its resonant frequency as less current is drawn from the device — with the drawback of limiting the capabilities of the tactile experience. Such ad hoc re-mapping of tactile information to the capabilities of an actuator is thought of an important requirement for a tactile framework, and is an integral part of the adaptive resynthesis from the parametric data model conceptualized in this work.

Lacking a universal data model for tactile stimuli

To enable the possibility for tactile experience designers to author (audio-)tactile assets *once* and ship them on a variety of devices (f.e., smartphones, video game consoles, desktop computers) it is important to define a platform-agnostic data model and file format. This is an important feature because software applications and media assets are usually made available on a variety of devices. Similar to how audio and video assets can be played back on various platforms, without having to author for each individual platform, the same should be true for tactile assets. The tactile data model would need to include all information to adequately resynthesize a tactile stimulus on the device by either interfacing with an existing tactile API or by integrating a standardized resynthesis (decoding) engine. The content of the tactile data model and resynthesis process could be inspired by the parametric resynthesis methods discussed in the section on previous work on audio-tactile translation. Using a parametric method derived from these studies would ensure the desired audio-tactile integrity discussed in the section on natural audio-tactile events.

An added benefit of a parametric encoding of the tactile data is, that the stimulus can more easily be adapted to the device specific constraints. As discussed above, for most applications the range of the stimulus frequency is bound to the resonant characteristics and dynamic range provided by the actuator. This means the boundaries of the reproducible frequency range of such systems is limited to the capabilities of the actuator that is built into the device. This illustrates the need for better hardware solutions or motor control to adequately recreate the intended stimuli despite these limitations. Strategies to linearize the frequency response of tactile actuators are fairly well understood [25, 8], but could be insufficient due to the drastic resonant characteristics observed in current actuator models (see Figure 2). For this reason an parametric representation, which changes the characteristics of the *resynthesis* based on *actuator specific information* derived from standardized measurements (see Figure 3), is proposed to be a solution to serve both legacy and future actuation technologies. By informing the decoder (resynthesis) about the actuators capabilities an "adapted" signal could be synthesized, fitting the wide- or narrow-band characteristics and dynamic range of the actuator and therefore allowing the best possible stimulus to be reproduced. This process can be seen outlined in Figure 3: The "Actuator Specific Configuration" file is embedded on the receiving device and dictates the resynthesis procedure for the stimulus in accordance with the capabilities of the integrated actuator. This could mean that the signal is band-limited, frequency shifted or otherwise filtered to ensure the best possible reproduction of the intended stimulus to be reproduced. When a parametric representation of the intended tactile stimulus is provided (as seen in previous audio-tactile translation methods) re-mapping frequency components and filtering becomes a comparably trivial task. Conceptually, this process can be thought of as "adaptive degradation", which is common for applications using video graphics that need to adapt to the capabilities of the graphics processing unit (GPU) on the reproducing system, thus lowering the resolution of the displayed graphics when necessary.

To conclude, the definition of a ubiquitous file format for vibrotactile stimuli is important and should allow for technology- and platform agnostic encoding. During the reconstruction (decoding) of the stimulus signal the technology specific characteristics need to be taken into account. To achieve the best reproduction of a stimulus the information on the actuation technology would therefore be required to be embedded on the device intended to resynthesize (decode) the stimulus.

Summarizing the requirements for a vibrotactile pipeline

From the challenges and proposed solutions described in the sections above we can conceptualize an idealized stack for tactile design, transmission and reproduction, as illustrated in Figure 3. The flowchart illustrates the three main subjects involved in the process (illustrated as red ovals), namely the tactile experience *designer*, the *product owner* (i.e., hardware manufacturer) and the *recipient* — which can be thought of as the end user.

The top part of the flowchart illustrates the workflow for a designer to curate vibrotactile assets, using existing audio assets as a starting point. The audio asset undergoes an analysis process (f.e., parametric signal decomposition) to encode the relevant tactile information. The resulting tactile data model enables authoring and editing on a ubiquitous format by specialized authoring and editing tools. The tactile information is

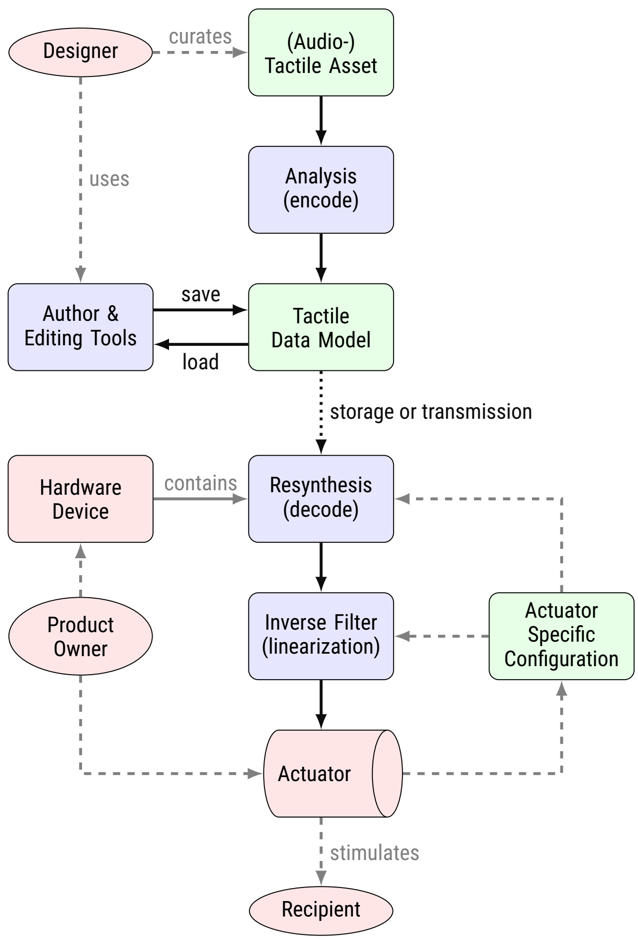


Figure 3: Flowchart of a proposed hardware integration and tactile stimuli design workflow. Subjects involved in the process are highlighted in the red ovals. Blue rectangles illustrate a process in the information flow. Green rectangles illustrate data files (i.e. information). The actuator hardware is illustrated as a cylinder.

contained in a tactile data model and can be stored until it is called or transmitted to a device for reproduction.

The bottom part of the flowchart illustrates the receiving side of the tactile delivery stack. The *product owner* is used as an umbrella term for the decision makers creating a new hardware product, such as a smartphone or game controller. They are in charge of selecting an adequate actuation technology for the vibrotactile capabilities of their device. Next, they should ideally be guided by standardized measurements and procedures to make sure the relevant information on the actuation technology (i.e. the *actuator specific configuration*) is made available to the *resynthesis* engine (decoding process) and linearization process (*inverse filter*) to ensure an accurate reproduction of the stimulus. This process ensures the best possible experience for the *recipient* (end user), while matching the intent of the *designer* as close as possible within the limits of the chosen tactile technology.

THE APPLE COREHAPTICS API

An example for a general-purpose tactile stimuli framework (including storage, transmission and reproduction) can be found integrated in current smartphone operating systems. For Android devices the need for a standardized framework is big, as every manufacturer shipping their devices with Android uses various tactile actuators and often creates a custom Android version that deviates in functionality from the stock version. The stock Android version provides a rudimentary API to interface with tactile actuator hardware using a "`VibrationEffect`" class. Next to a few pre-defined tactile patterns (f.e., click, double-click) it is possible to create custom effects by a sequence of time-amplitude pairs using the `createWaveform` call. There is no support in adjusting the frequency of the stimulus provided though. The newer Android R version promises a `VibrationEffect.Composition` class, which supports a composition of predefined so-called "primitives". In both cases, an audio-tactile framework would require the tactile signal to be approximated using these pre-defined assets, which does not conform with the parametric representations we could observe in the audio-tactile methods described above. Due to the variability in the Android space, the concept of authoring vibrotactile assets once and shipping them to all Android devices would be complicated due to the custom Android versions and varying implementations of the vibration-effect classes of the stock Android API. For this reason, the following analysis primarily focuses on the CoreHaptics API by Apple, because in their implementation there are no deviating versions and all devices supporting CoreHaptics share the same API and tactile data model (AHAP), which is in agreement with the requirements gathered in the sections above.

The `CoreHaptics` stack consists of a software API, that can be integrated in iOS applications to provide an interface for the "Taptic engine" actuators embedded in a range of iOS devices, such as the Apple Watch; the late 2014 12" and 13" MacBook or newer; and the iPhone 6s or newer iPhone models. The CoreHaptics API is mostly a black box system that can not be fully understood by reading the documentation only. The resulting vibrational output of API calls can better be understood by conducting "abstracted" system identification measurements, as the API uses a proprietary input file format called AHAP (Apple Haptic and Audio Pattern). The implementation allows a user to define a haptic (i.e. vibrotactile) pattern in conjunction with an audio file – allowing both events to be played synchronously.

The AHAP file format is composed of both *continuous* and *transient* components. The continuous type `hapticContinuous` is initialized with a fixed duration of the tactile event and is further specified by an initial "intensity" and "sharpness" value. Using a so-called `parameterCurve` it is then possible to modulate the intensity and sharpness values over time. Both parameter types are defined as a floating point and range from 0 to 1 and can be set for the entire duration of the tactile event. This functionality can be thought of as a parameter automation. The parameter curve applies the same way for both transient and continuous events. The initialization of the tactile event and the automation are stored in the AHAP container in a descriptive JSON style file format. An event can then be called by the CoreHaptics API to be played back by the Taptic Engine of the iOS device.

EXPERIMENTAL MEASUREMENTS

The `intensity` value described above can be thought of as a parameter to control the amplitude of the resulting tactile stimulus, while the `sharpness` value modulates the frequency (cf. [36]). These insights were gained through experimental measurements using a test pattern[2] in the AHAP file format and measuring the acceleration response of an iPhone 8 as seen in Figure 4. The first nine steps in the test pattern

[2] AHAP test pattern used for the measurements in Figure 4 `http://myjson.com/6ey2r`

generate a continuous event with a sharpness of 0.5 and a duration of 0.9 seconds — while the intensity is increased for each step. The second part of the test pattern is composed of a 5 second long frequency sweep — modulating the sharpness value from 0 to 1 over the entire duration. The measurements were conducted by mounting the iPhone 8 in a test jig and attaching a ADXL325 accelerometer to the device case in close proximity to the Taptic Engine. The signal was recorded using a MOTU 624 audio interface, which was previously calibrated using a Siglent SDG1010 signal generator and a Rigol DS4014 oscilloscope.

Through the measurement results seen in Figure 4 we can observe that the intensity (i.e. amplitude) is scaled logarithmically. For a designer wanting linear control over the amplitude this means the values in the parameter curve need to be pre-squared to achieve the desired linear result (see Figure 4a). Furthermore, we can be observe that the sharpness value from 0 to 1 map to a corresponding frequency range from 80 to 230 Hz. It also seems like the actuator is not linearized, which results in the difference in the acceleration magnitude of more than 12 dB between 80 to 160 Hz — even though the intensity value wasn't changed (see Figure 4b). Through this observation it is suspected that the Taptic Engine in the iPhone 8 has a resonance frequency of 160 Hz, which is reached with a sharpness value of 0.7. The observation on the resonance frequency could be verified by driving the Taptic engine directly without the use of the API, as seen in Figure 2.

When generating an AHAP pattern it is important to note that there are absolute and temporal limits to the amount of parameter curve values that can be defined. For each parameter curve only a set of 16 points can be used. This limitation can be bypassed by using multiple parameter curves in succession. The maximum duration for a tactile event has been shown to be 30 seconds by measurements conducted with a similar test pattern file. A limitation on the maximum update rate, i.e. a limit of parameter automation events per second has been observed but has thus far not been identified. Further experimental tests have indicated that transient events seem to have a fixed predefined wave shape. This can lead to phase cancellation during playback when triggering a transient event during a continuous event. It has been observed that this can result in an undesired "ducking" effect of the vibration instead of feeling the transient which was set to that position.

It is also important to note that for a single AHAP file the automation of intensity and sharpness values by the parameter curve influences both the intensity and sharpness of all contained continuous and transient events. There is a difference in the way the modulation works for intensity and sharpness though. For example an initial *event* intensity i_e gets multiplied with the current `parameterCurve` value i_{pc}, so that $i_e \cdot i_{pc} = p_r$, resulting in a *multiplicative* modulation p_r. The sharpness values are modulated *additively* though, which means that an initial sharpness value s_e is modulated by the current `parameterCurve` value s_{pc} by $s_e + s_{pc} = s_r$. This requires the range of the parameter curve values for intensity to be on the interval $i_{pc} \in [0.0, 1.0]$ and the values for sharpness to be on the interval $s_{pc} \in [-1.0, 1.0]$. The initial values for both continuous and transient events are set on the interval $[0.0, 1.0]$ for both the intensity and sharpness. This is important to note, as it can happen that a parameter curve automation completely "mutes" a transient event — thus resulting in an undesired result.

Accompanying audio-tactile design tools

As the CoreHaptics API is limited to a combination of a single frequency continuous or a transient event the complexity of the tactile stimulus is inherently limited by the bounds of the parameter curve values and the event calls. To approximately map an audio asset to a tactile stimulus we need to estimate both the intensity and sharpness of the continuous and transient events over time based on the audio source material. Manually creating a tactile stimulus in the AHAP file format by filling in the JSON container is a tedious task. To make this task easier, various design tools have been developed. Next to a more user friendly environment to design tactile stimuli for the AHAP file format by using a graphical user interface (GUI), some of the following tools integrate an audio-tactile translation approach.

The software "Haptrix"[3] comes as a standalone application for OSX and allows the user to sync to a local iOS device to play back the tactile stimuli. It features a Attack-Decay-Sustain-Release (ADSR) envelope model that most audio designers are probably familiar with to make the design of the paramter curve easier. It further allows for an automatic transient detection when loading an audio asset into the editor and enables to set the transient events in the AHAP file accordingly.

A more sophisticated audio-tactile focused approach comes from the company Lofelt[4]. This browser-based application allows the user to load an audio file and the process returns an approximate tactile stimuli in the AHAP file format by automatically detecting transients and retrieving an envelope for the intensity and sharpness curves. The browser session can be synced to an accompanying iOS application by scanning a QR code. By establishing this link it is possible to feel the tactile stimulus that is currently open in the editor of the composer application. Through the combination of the composer tool and the CoreHaptics API a design workflow reaching from the intended design to the reproduction can be realized — with the added benefit that the reproduction during the design mirrors the exact reproduction on the recipient (user) side. Although not perfect, CoreHaptics could serve as a blueprint for a standardized general purpose framework that could work across various platforms. To ensure a true wide-band vibrotactile stimuli display the tactile data model and resynthesis would need to be expanded though.

Using the design tools described above accelerates the design of a tactile stimuli in the AHAP format greatly. Application developers, researchers, designers or other users can utilize the large variety of already existing audio assets and create a matching stimulus — at least within the limitations of the CoreHaptics framework. How useful the CoreHaptics API could become for developers or researchers still needs to be decided for each individual case as the capabilities of the API need to match the requirements of the case study or application. Slightly complicating the matter for researchers is the fact, that some iOS devices allegedly use different model of the Taptic Engine (tactile actuator) which needs to be taken into account when generating the stimulus on a device basis. How strong the differences of the resulting stimuli between various Taptic engines are, and if these differences are perceptually significant needs to be analysed further.

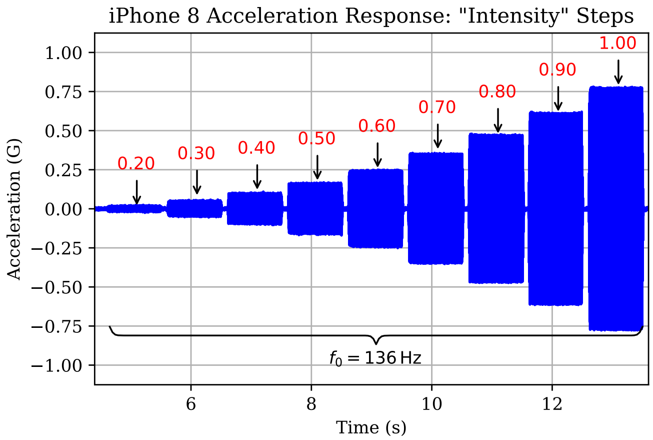
To conclude, the CoreHaptics API provides an interesting proposal on how to integrate a device-agnostic API for vibrotactile assets. From the analysis we can summarize that it does not entirely conform to the desired wide-band functionality when customizing tactile patterns, but provides a big step away from generic buzzing vibrations and overly complicated API implementations found in other frameworks. Furthermore, the linearization and actuator specific driving methods could be improved to provide a closer match to the intended stimuli defined by the AHAP format.

Conclusion

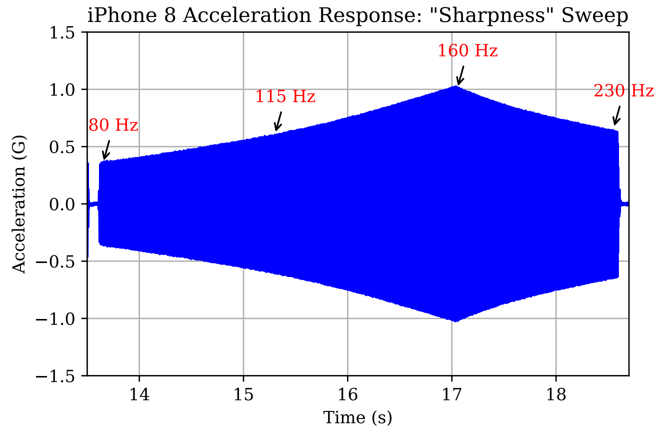
In this work various challenges in the end-to-end transmission of tactile stimuli — from design to reproduction — have been discussed. Key requirements and important stakeholders required for a standardized framework were identified to ensure high fidelity stimuli reproduction across various platforms and tactile technologies. Given the comparably low resolution of the tactile sense a coarse parametric signal representation appears to be a feasible option for a tactile data model (and

[3] Haptrix on the Mac App Store
`https://apps.apple.com/us/app/haptrix/id887185157`

[4] Lofelt Composer for audio-tactile design
`https://composer.lofelt.com/`



(a) Intensity Steps



(b) Sharpness Sweep

Figure 4: Plot 4a displays the acceleration response of intensity value steps on an iPhone 8 playing back the AHAP test pattern file (see Footnote 2). The intensity value for each step is displayed in red text. The sharpness value was set to 0.5, which corresponds to a frequency of 136 Hz on the iPhone 8's Taptic engine. Plot 4b displays a 5 second sweep of the sharpness value from 0 to 1. The momentary frequency is shown in the red annotation. A resonance around the sharpness value of 0.7 with 160 Hz can be observed. The intensity value was set to 1.0 throughout the entire sweep. When comparing the acceleration response to the data displayed in Figure 2, it becomes apparent that some measures of linearization have been taken when driving the actuator through the API.

file format). Next to the benefits in data compression, a parametric representation could further enable the capability of an adaptive resynthesis feature that ensures the best possible stimulus reproduction within the limitations of the tactile technology at hand. To enable an adaptive resynthesis feature, characteristics of the tactile display technology would ideally be present during the resynthesis (decoding) process of the tactile stimulus. The required characteristics of the tactile display technology would need to be captured using standardized measurement procedures to identify key metrics, such as the available bandwidth, resonance frequency and dynamic range of the tactile display technology. Further measurement procedures and quality metrics, as the ones used for loudspeakers [6], could serve as a blueprint to aid the development of standardized tests and metrics for tactile actuators. So far proposals for standardized tests and metrics mainly come from industry players [23, 15] and the ISO standard [17, 16, 18]. A standard set of metrics could pave the way for meaningful quality levels of actuator technologies and further improve the quality of available information around tactile devices — thus informing decision processes depending on the vibrotactile systems capabilities. One use case being the adaptive characteristics of a codec, supporting both legacy and future vibrotactile technologies. A prototype implementation of the conceptual framework presented in this work, including an audio-tactile encoding stage, a parametric data model and an adaptive resynthesis system is planned to be presented in a forthcoming publication.

Lastly, as an exemplary use case, the CoreHaptics API was analysed to get a better understanding of the current state of the most advanced and widespread solution to market. Compared to the dull and inaccessible buzzing vibration patterns of the infamous 80s pagers and early mobile phones, great improvements have been made on the current state of tactile reproduction systems on the market. Further facilitating easy content integration and high fidelity reproduction of tactile stimuli will open up novel forms of creative expression, increase accessibility and foster medical applications for many people, as it opens up the possibility to address the potential of a mostly untapped sense.

Acknowledgements

The authors would like to thank the company Lofelt GmbH for providing hardware, test equipment and office space to enable this work — as well as all thought provoking and inspiring ideas coming from the team.

A special thanks goes to Andrew Lazdins for modifying the smartphones, Sreedhara Kolindala for conducting the measurements and providing the raw data, and to João Freire for the measurements conducted on the iPhone 8.

References

[1] M. E. Altinsoy and S. Merchel. BRTF (body-related transfer function) and whole-body vibration reproduction systems. In *Audio Engineering Society Convention 130*, May 2011.

[2] E. T. Auer Jr, L. E. Bernstein, W. Sungkarat, and M. Singh. Vibrotactile activation of the auditory cortices in deaf versus hearing adults. *Neuroreport*, 18(7):645, 2007.

[3] D. Birnbaum and M. M. Wanderley. A systematic approach to musical vibrotactile feedback. In *ICMC*, pages 397–404, 2007.

[4] S. J. Bolanowski Jr, G. A. Gescheider, R. T. Verrillo, and C. M. Checkosky. Four channels mediate the mechanical aspects of touch. *The Journal of the Acoustical society of America*, 84:1680–1694, 1988.

[5] D. Boyd, D. Gould, M. Hutson, M. Klawunn, D. H. Laidlaw, J. Chan-Norris, J. Potter, and H. Rohde. Depth cues in virtual reality and real world: Understanding individual differences in depth perception by studying shape-from-shading and motion parallax, 2000.

[6] BRITISH STANDARD. *IEC 60268-5:2003 - Sound system equipment - part 5: loudspeakers*. International Electrotechnical Commission, 2003.

[7] P. Brooks and B. J. Frost. Evaluation of a tactile vocoder for word recognition. *The Journal of the Acoustical Society of America*, 74(1):34–39, 1983.

[8] A. T. Bukkapatnam, P. Depalle, and M. Wanderley. Autoregressive parameter estimation for equalizing vibrotactile systems. In *Proceedings of the 2019 International Workshop on Haptic and Audio Interaction Design (HAID). Lille, France.*, 2019.

[9] F. Fontana, H. Järveläinen, S. Papetti, F. Avanzini, G. Klauer, L. Malavolta, C. di Musica, and C. Pollini. Rendering and subjective evaluation of real vs. synthetic vibrotactile cues on a digital piano keyboard. In *Proceedings of the Sound and Music Computing Conference. Maynooth, Ireland: SMC*, 2015.

[10] O. Franzén and J. Nordmark. Vibrotactile frequency discrimination. *Perception & Psychophysics*, 17(5):480–484, 1975.

[11] G. A. Gescheider, S. J. Bolanowski, and S. K. Chatterton. Temporal gap detection in tactile channels. *Somatosensory & motor research*, 20(3-4):239–247, 2003.

[12] H. Gillmeister and M. Eimer. Tactile enhancement of auditory detection and perceived loudness. *Brain research*, 1160:58–68, 2007.

[13] G. D. Goff. Differential discrimination of frequency of cutaneous mechanical vibration. *Journal of experimental psychology*, 74(2p1):294, 1967.

[14] S. Guest, C. Catmur, D. Lloyd, and C. Spence. Audiotactile interactions in roughness perception. *Experimental Brain Research*, 146(2):161–171, 2002.

[15] Immersion Corporation. Immersion: The haptic stack – hardware layer (blog post). https://www.immersion.com/the-haptic-stack-hardware-layer/, 2020. 25.03.2020.

[16] ISO. *Ergonomics of human-system interaction – Part 920: Guidance on tactile and haptic interactions*. International Organization for Standardization, 2009.

[17] ISO. *Ergonomics of human-system interaction – Part 910: Framework for tactile and haptic interaction*. International Organization for Standardization, 2011.

[18] ISO. *Ergonomics of human-system interaction – Part 940: Evaluation of tactile and haptic interactions*. International Organization for Standardization, 2017.

[19] V. Jousmäki and R. Hari. Parchment-skin illusion: sound-biased touch. *Current biology*, 8(6):R190–R191, 1998.

[20] M. Karam and D. I. Fels. Designing a model human cochlea: Issues and challenges in crossmodal audio-haptic displays. In *Proceedings of the 2008 Ambi-Sys workshop on Haptic user interfaces in ambient media systems*, pages 1–9, 2008.

[21] C. Kayser, C. I. Petkov, M. Augath, and N. K. Logothetis. Integration of touch and sound in auditory cortex. *Neuron*, 48(2):373–384, 2005.

[22] A. Lerch. *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press, 2012.

[23] Lofelt GmbH. Vt-1: A specification proposal for realistic vibrotactile feedback. in *2019 Smart Haptics*. https://lofelt.com/resources, 2019. 21.02.2020.

[24] M. T. Marshall and M. M. Wanderley. Vibrotactile feedback in digital musical instruments. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 226–229, 2006.

[25] W. McMahan and K. J. Kuchenbecker. Dynamic modeling and control of voice-coil actuators for high-fidelity display of haptic vibrations. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 115–122. IEEE, 2014.

[26] S. Merchel and M. E. Altinsoy. Auditory-tactile music perception. In *Proceedings of Meetings on Acoustics ICA2013*, volume 19, page 015030. Acoustical Society of America, 2013.

[27] S. Merchel and M. E. Altinsoy. Music-induced vibrations in a concert hall and a church. *Archives of Acoustics*, 38(1):13–18, 2013.

[28] S. Merchel and M. E. Altinsoy. The influence of vibrations on musical experience. *J. Audio Eng. Soc*, 62(4):220–234, 2014.

[29] S. Merchel, M. E. Altinsoy, and M. Stamm. Just-noticeable frequency differences for whole-body vibrations. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, pages 2234–2239. Institute of Noise Control Engineering, 2011.

[30] D. A. Norman. *The psychology of everyday things.* Basic books, 1988.

[31] M. Papadogianni-Kouranti. Auditive and audiotactile music perception of cochlear implant users. *A Thesis presented for the degree of Master of Science*, pages 1–28, 2014.

[32] S. Papetti, S. Schiesser, and M. Fröhlich. Multi-point vibrotactile feedback for an expressive musical interface. In *NIME*, pages 235–240, 2015.

[33] S. Rimell, D. M. Howard, A. M. Tyrrell, R. Kirk, and A. Hunt. Cymatic. restoring the physical manifestation of digital sound using haptic interfaces to control a new computer based musical instrument. In *ICMC*, 2002.

[34] C. Saitis, C. Fritz, and G. Scavone. Sounds like melted chocolate: How musicians conceptualize violin sound richness. In *2019 International Symposium on Music Acoustics (ISMA)*, 2019.

[35] C. Saitis, H. Järveläinen, and C. Fritz. The role of haptic cues in musical instrument quality perception. In *Musical haptics*, pages 73–93. Springer, Cham, 2018.

[36] C. Saitis and S. Weinzierl. The semantics of timbre. In *Timbre: Acoustics, perception, and cognition*, pages 119–149. Springer, 2019.

[37] B. E. Stein and M. A. Meredith. *The merging of the senses.* The MIT Press, 1993.

[38] A. Tajadura-Jiménez, A. Väljamäe, I. Toshima, T. Kimura, M. Tsakiris, and N. Kitagawa. Action sounds recalibrate perceived tactile distance. *Current Biology*, 22(13):R516–R517, 2012.

[39] J. B. Van Erp, K.-U. Kyung, S. Kassner, J. Carter, S. Brewster, G. Weber, and I. Andrew. Setting the standards for haptic and tactile interactions: Iso's work. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 353–358. Springer, 2010.

[40] J. M. Yau, J. B. Olenczak, J. F. Dammann, and S. J. Bensmaia. Temporal frequency channels are linked across audition and touch. *Current biology*, 19(7):561–566, 2009.