Doctoral Thesis

Attachable/Detachable Vibrotactile Modules and their Applications

Gunhyuk Park (박건혁)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2017

탈부착이 가능한 진동 피드백 장치와 그 응용

Attachable/Detachable Vibrotactile Modules and their Applications

Attachable/Detachable Vibrotactile Modules and their Applications

by

Gunhyuk Park

Department of Computer Science and Engineering

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A thesis submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science and Engineering

Pohang, Korea
December 19, 2016
Approved by

Seungmoon Choi

Seungmoon Choi, Academic Advisor

Attachable/Detachable Vibrotactile Modules and their Applications

Gunhyuk Park

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH.

12/19/2016

Committee Chair	최 승 문	(Seal)
Member	한 성 호	(Seal)
Member	이 승 용	(Seal)
Member	한 보 형	(Seal)
Member	김 상 연	(Seal)

DCSE 20075435 박건혁 Gunhyuk Park, Attachable/Detachable Vibrotactile Modules and their Applications. 탈부착이 가능한 진동 피드백 장치와 그 응용, Department of Computer Science and Engineering, 2017, 111P, Advisor: Seungmoon Choi. Text in English

Abstract

With the increase interaction with many devices, multimodal displays can help users to acquire various information. Although there are literally thousands of researches of haptic interfaces, a small number of them are commercialized because of the high cost of embedding hardware to multimodal displays. The goal of this study is to develop an attachable and detachable haptic modules, haptic enchanters, to provide localized vibrations on user-desired locations with high information transfer (IT) rates.

Haptic enchanters consisted of an attachment part, a vibration isolation layer (optional), and a vibration actuator. Nine prototypes of box-type and ring-type haptic enchanters were developed considering target surface of a rigid flat mobile phone and a human finger and three of them did not included the vibration isolation layer. The vibration isolation layer was selected through an extensive measurement process, and a silicone layer of hardness 0030, softner ratio of 15 %, and 3 mm of thickness showed over 90 percent of vibration attenuation with a moderate gripping. The ring-type prototypes directly contact to human fingers and do not need the isolation layer.

Perceptual qualities using haptic enchanters are estimated through three user studies in Chapter 3 and 4. As a performance measure of a platform for information transmission, information transfer (IT) is used in two user studies in Chapter 3 and 4. In the representative configurations of PHONE and RING, spatiotemporal vibration sequences using 2, 3, and 4 haptic enchanters provided the high capacity of information transfer (4.55-7.06 bits). For the stationary spatial patterns, Illusory 2D phantom sensations induced by using four haptic enchanters also showed information transfer of 1.89-2.53 bits for 3 by 3, 4 by 4, and 5

by 5 virtual grids. The result implied that 2D phantom sensations had a potential as a secondary information channel, so the illusory sensation resolution was measured more precisely. The perceptual resolution of using 2D phantom sensations were 5 by 5 and 5 by 4 grids for PHONE-LINEAR and PHONE-LOGARITHM and 6 by 5 and 6 by 4 grids for RING-LINEAR and RING-LOGARITHM. These results instantiate the promise of haptic enchanters as effective and convenient communication accessories.

An authoring tools was required to utilize haptic enchanters. First, a software library, PhysVib, was developed on the mobile platform by extending an open-source physics engine in a multi-rate rendering architecture for automatic vibrotactile feedback upon collision events. PhysVib runs concurrently with a physics engine at a low update rate and generates vibrotactile feedback commands at a high update rate based on the simulation results of the physics engine using an exponentially-decaying sinusoidal model. A user study showed that this vibration model is more appropriate to our purpose in terms of perceptual quality than more complex models based on sound synthesis. The perceptual performance of PhysVib was estimated by comparing eight vibrotactile rendering methods. Experimental results suggested that PhysVib enables more realistic vibrotactile feedback than the other methods as to perceived similarity to the visual events. PhysVib is an effective solution for providing physically plausible vibrotactile responses while reducing application development time to great extent. Multiple haptic enchanters are also available by extending PhysVib to support multi-channel vibrotactile signals using phantom sensation rendering methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Goal of Research

Haptics research has nurtured the development of literally thousands of relevant interfaces, ranging from the early effort in 1920s for speech-to-tactile conversion [30] to the Taptic Engine in Apple watches released in 2015. Haptic interfaces can sense the user's movement and provide appropriate stimuli for feedback in many forms, e.g., position, pressure, vibration, force, skin stretch, friction, texture, heat, and motion [34]. This diversity has allowed haptic interfaces to be utilized in a great number of applications for improved and enriched interaction [20], such as information communication [29], multimedia [26], data perceptualization [74], to name a few, in real, virtual [92], or augmented environment [45].

Despite such prolonged and focused research endeavors, haptic interfaces still await adoption in a wider range of applications and consumer products, other than in gaming devices, mobile phones, and cars. This is largely due to the fact that users' perceived benefit does not considerably exceed the cost required to add haptic feedback into existing products. In most cases, haptic feedback is used sparingly and purposefully to maximize its intended effects while not causing fatigue to the users, sometimes playing a secondary and ambient role in interaction. This is in contrast to visual and sound displays that are generally used as the main displays.

Many approaches to improve upon the situation have been investigated over the years, and my best idea for that has been *haptic enchanters*. Haptic enchanters refer to small modules that can be attached to ordinary devices and objects to endow them with the ability of providing programmable haptic stimuli, thereby transforming them to haptic-enabled interfaces. Haptic enchanters should also be easily detachable to preserve the original usage of the objects. The name is based on an analogy to enchanting items in computer games.

Developing a vibrotactile stimuli design tool is equally essential meanwhile the haptic enchanter system is required to be developed and evaluated. Designing vibrotactile stimuli appropriate to a system is a challenging and time consuming task and many researchers have developed software tools such as graphical authoring tools, automatic conversion from sound to vibrotactile effects, and automatic generation of vibrotactile effects by using a physics engine. These previous software effectively improve a performance, however, they were appropriate to haptic systems using the fixed number and the fixed location of actuators. This thesis extends a standard open-source 2D physics engine to render vibrotactile feedback using the collision information and name it PhysVib: a software library on the mobile platform extending an open-source physics engine for automatic vibrotactile synthesis upon collision events.

In this thesis, main contributions are threefold: 1) development of haptic enchanter prototypes and estimation of their perceptual performance, 2) an automated vibrotactile feedback library for haptic enchanters, and 3) integration of the haptic enchanter prototypes and the library for spatiotemporal feedback. The rest of this introduction provides contributions and an overview of this paper.

1.2 Contributions

The major contributions of this study are summarized as follows:

- Development of prototype haptic modules that are attachable to and detachable from a rigid and flat surface or a human finger,

- Estimation of information capacity using haptic enchanters,

- Measurement of a spatial resolution of 2D stationary phantom sensations using haptic enchanters,
- Verification of feasibility of a haptic enchanter by applying it to the physically-plausible vibrotactile feedback library, PhysVib, and
- Integration of haptic enchanters and PhysVib.

1.3 Organization

In Chapter 2, backgrounds with respect to perceptual characteristic of vibrotactile feedback, single and multiple actuator system, and vibrotactile feedback authoring tools are presented. Structure of haptic enchanters and estimated performance on intra-hand and inter-hand configurations are described in Chapter 3. In Chapter 5, physically-plausible vibrotactile feedback library on collision events, PhysVib, is described for its realistic vibrotactile feedback using a single haptic enchanter. The haptic enchanters and the PhysVib are then integrated to build a customizable haptic feedback system in Chapter 6. We conclude the current study in Chapter 7.

Chapter 2

Background

2.1 Perceptual Characteristic of Vibrotactile Feedback

The perceived intensity and subjective perception of vibration are affected by many factors including vibration amplitude, frequency, contacted body site, contact area, stimulus duration, vibration direction, and stimulator weight [47, 83]. Some factors affect little on a mobile vibrotactile system, for example, mobile platforms usually interact with users via hands and body site is fixed to the human hands. Also, increasing contact area reduces the detection threshold of vibration, but it saturates when the contact size exceeds 2.9 cm^2 [93] where most of mobile devices have larger contact area [83]. A previous study investigated perceived intensities varied by stimulator weight and vibration direction, and the weight was not that significant in 40-g difference, while the vibration direction affected the perceived intensity [42].

The relationship between a physical vibrotactile stimulus and the perceived intensity has been the fundamental topic in haptics. The perceived intensity increases as the vibration amplitude increases starting from a detection threshold as a reference zero intensity [31], where the detection threshold refers the smallest signal intensity that can be guaranteed its presence. The detection thresholds of vibrotactile stimuli follows a U-shaped curved function of frequency, that have the lowest threshold between 200 and 300 Hz with a displacement unit of mm [47]. A stimulus duration requires over 1-s in psychophysical studies for a perceived intensity measurement to avoid the temporal summation of the PC channel, but shorter durations are used in vibrotactile feedback design to form a vibrotactile pattern [10, 67].

Vibrotactile pattern design requires perceptually well-discriminable parameters to form a pattern, and Brewster et al. suggested the Tacton using design parameters with frequency, amplitude, waveform, duration, rhythm, body location, and spatiotemporal patterns [10, 35], and MacLean et al. proposed the Haptic Icon with a similar concept design [67, 76]. In a view of perceptual design, the Tacton proposed parameter selecting rules by difference threshold and just-noticeable-difference (JND), and the Haptic Icon showed perceptual distances of designed patterns using a Multi-Dimensional Scaling (MDS) method. Users can easily discriminate if a pair of vibrations has larger perceptual distance than other pairs, thus ideal object of the vibrotactile design is maximizing all perceptual distances between vibrotactile feedback pairs.

2.2 Tactile Feedback System

Researchers have developed and enhanced various vibrotactile feedback systems to provide rich information to users [29]. Begun from the sensory substitution for the disabled, these purposes have been accomplished by providing information via the tactile sensory channel, and it requires well-defined coding scheme by taking care of the target application, vibrotactile perception, and users. To deliver information, utilizing a single actuator with various vibrotactile parameters and multiple-actuator system were two main approaches in a view of system design.

2.2.1 Single-actuator System

A single-actuator system contains only one actuator on the device, and users have to feel or discriminate vibrotactile stimuli by the temporal differences. A smart phone is the most common haptic device that utilizes one actuator to vibrate the whole body to deliver information. Many researchers utilized one haptic actuator were proposed by varying temporal vibration parameters including amplitude, frequency, waveform, rhythm [66], but most of them were unable to exceed 3 bits of information transfer. This single-actuator system is affordable to enhance user experiences or alarm users, but not good to deliver various and fertile information because the human vibrotactile perception of temporal factors is relatively poor [102].

2.2.2 Multi-actuator System

In haptics, various multi-actuator system has been developed and researchers carefully designed vibration patterns to increase information channel capacity for a higher information transfer. First of all, multiple tactile stimuli can be applied to different body sites simultaneously (spatial patterns) or to the same location sequentially (temporal patterns), or as spatiotemporal patterns by combining the two. Spatial patterns are generally intuitive for information transmission through the tactile sense [12, 20]. A popular form factor for that is to use an array of tactile actuators, e.g., on the fingertip [65], the back [43], or even the whole body [62]. Tactors can also be embedded into wearables such as wristbands [61], wristwatches [77], and belts [101].

Sensory substitution is another good application that uses multiple actuators to substitute other sensory channels for the disabled people. Among the sensory modalities, the visual information had been the main target to be converted to tactile information using spatial patterns. One of those systems is ”tactile vision substitution systems”(TVSS), and Bach-Y-Rita et al. transformed an image captured by a TV camera to 20 by 20 vibration actuators attached on the chair [6, 22, 103]. Using TVSS, subjects could perceive horizontal, vertical, and diagonal lines immediately and experienced subjects could perceive even people’s face. The Kinotact system, which was developed by Craig et al., had a similar actuation system with 10 by 10 vibrators and converted letters to vibrations [25]. Linvill and Bliss et al. developed one of the most successful devices, Optacon (OPtical-to-TActile CONverter), that helps the blinded people read printed materials. By scanning with a photocell camera, the device converts the scanned image to tactile feedback using a piezoelectric bimorphing display on a fingertip [63, 8]. Substitution of auditory was another target utilizing the tactile feedback system using spatiotemporal patterns. Nature of sound is a transferred vibration of an object via the air, and researchers paid attention to deliver the sounds to the hearing impaired people. Queen's University tactile vocoder, a 16-channel device with an array of magnetic solenoid transducers arranged linearly along the forearm, was the representative system for auditory substitution [11]. Tactaid series reduced transducer channels than Queen's device and showed similar performance.

While sensory substitution used patterns to convert visual and auditory information to tactile stimuli, there have been another approaches to provide abstract information using tactile stimuli. For example, multiple non-localized actuators were attached on a phone-shaped rigid body and provided various vibration patterns [49, 51, 87, 106], directly attached on human skin [19, 61], or localized using the vibration isolation material [81]. These systems mainly aimed to provide many information and helped to give intuitive spatial cues. Multiple-actuator systems reported over 4 bits with tactile localization [38, 61] or 3.7 bits utilizing superpositioned vibrations without localization [88]. Vibration contactors of T-Hive are localized, however, the device is too large to be applied in the mobile platforms.

2.2.3 Vibration Isolation

In this thesis, the main difficulty to develop haptic enchanters is that the contactor of the tactile system is not limited to the enchanters. The minimum requirement was providing spatially-localized tactile feedback in *rigid* devices, e.g., smartphones, since vibrations propagate through the device's body. Without it, only temporal information coding is allowed, and this greatly impairs the information transmission capacity.

Vibration localization (isolation) requires an adequate damping mechanism. For instance, T-Hive adds a polyurethane layer between a spherical handle and actuators to stimulate the palmar side of the hand with 13 localized vibration panels [81]. T-Mobile applies the same idea to a mobile phone case that has a 4×3 array of tactors [105]. These two papers report good discrimination between the vibrating pads, but they do not specify vibration attenuation ratios. HoliBraille is a mobile device case presenting a 3×2 Braille array of vibration motors [71]. For its design, the authors tested the vibration attenuation performance of three materials, a sorbothane layer, a cork layer, and a steel spring, using 150-Hz vibrations. The former two were not effective, but the steel spring could attenuate vibration by 99%. However, it also exhibited significant nonlinearity shifting the vibration frequency from 150 Hz to 90 Hz, which is generally unacceptable for vibrotactile rendering. For the same purpose, SemFeel embeds five vibration motors to a thick sleeve made of sponge-like material attached to the back of a touchscreen device [106]. All these studies can still benefit from systematic studies that find best materials for vibration attenuation.

Haptic enchanters are designed to provide localized vibrotactile feedback to users, and it is natural to expect the pros of using them as the spatial resolution using concrete sensations or illusory sensations. To measure the information capacity of those two methods, information transfer is a well-known and generalized estimator.

2.2.4 Information Transfer

Information Transfer (IT) is frequently used in perceptual studies as a measure for quantifying the *channel capacity* of a human sensory modality. Since IT is context-free and task-independent, it is appropriate for comparing the fundamental information transmission capacities between different displays [98]. Experiments measuring the IT of a sensory channel should not require participants to remember the stimulus-response mapping. This is to estimate solely the communication capacity of the sensory channel, not interfered with the noise of human memory capacity.

Many studies reported the IT values of different haptic displays and stimulus sets. Using a single vibrotactile actuator, 1.76 bits of IT was measured with 8 stimuli that varied in amplitude, frequency, and pulse duration [5]. A higher value of IT, 2.5 bits, was observed with 12 key-click signals with different amplitudes, frequencies, and the number of pulses [18].

When multiple tactors are used with suitable localization, IT is generally higher. The IT values reported in the literature include: 1.99 bits with two 3×3 tactor arrays (18 in total) placed on the dorsal and volar sides of the wrist [19]; 1.90–2.49 bits with 8 tactile patterns using 4 voice coil actuators around the wrist [68]; 2.46 bits with 9 locations with a 3×3 motor array affixed to the palm [90]; 3.37 bits—27 tactile stimuli with different rhythms, roughnesses, and spatial locations using 3 voice coil actuators stimulating the forearm [13]; 4.28 bits—24 vibration patterns generated using 3 vibration motors in a wrist-band tactile display by changing their intensity, temporal pattern, starting point, and direction [61]; and 6.5 bits–120 stimuli using 4 finger configurations (thumb, index finger, middle finger, or all of them) and 30 waveforms with different frequencies and amplitudes [97].

When multiple tactors are used without stimulus isolation, illusory effects similar to apparent tactile motion can be rendered on a handheld device using appropriate algorithms [49, 86, 87, 88]. 3.70 bits of IT was reported from 32 vibrotactile flows rotating around the edges of a mobile device produced with 4 actuators [88]. The Buzzwear showed high information rate but vibrotactile feedbacks were indirect for the mobile application [61]. Edgeflow, however, showed 4.38 bits of information transfer by utilizing four actuators without tactor localization [87], with coincidence of visual and tactile feedback.

These results were estimated using non-localized vibrotactile stimuli, thus providing localized vibrations will improve the IT. In this thesis, localized vibrations using multiple actuators are rendered to a mobile phone and four fingers of human hands to estimate the extent to which the localizations improves the information transfer. The localization also provide the stronger illusory perception than previous multiple actuator systems, and the next subsection describes the illusion in detail.

2.2.5 Phantom Sensations

A phantom sensation refers to an *illusory* tactile sensation perceived midway between two or multiple distant vibrotactile stimulations [1, 43, 95]. The perceived location can be controlled by adjusting the amplitudes of the stimuli (amplitude inhibition) or their time gaps (temporal inhibition). Although this phenomenon was discovered in the 1970s [1], recent years have seen increasing interests in applying phantom sensations to haptic interaction. The main advantage is that only a small number of actuators are required.

Phantom sensations can be classified by temporal movement (stationary and dynamic), body site, and spatial dimension (1D or 2D). Stationary phantom sensations are also called funneling illusions, and dynamic phantom sensations are similar in their notion to apparent tactile motion and sensory saltation. Many prior studies addressed phantom sensations on different body sites [1, 7, 43, 95]. Our review below pertains to those for hands for their relevance to User Study 2.

1D stationary phantom sensations are generally robust perceptual illusions. For example, when 1D stationary phantom sensations were presented to a hand using three piezoelectric actuators that were 18 mm apart, the mean accuracy of recognizing nine illusory midway points was as high as 81.2% [105]. Research on 2D stationary phantom sensations is rare, except [53]. This work showed that 2D stationary phantom sensations rendered in a 5×7 virtual grid on a tablet (4 ERM actuators at the corners) grabbed with both hands resulted in an average percent correct (PC) score of 28.4%. Note that this case corresponds to "out of the body" since illusory sensations occur in the space between the two hands [53].

Dynamic phantom sensations have been studied more for their immediate applicability to movement rendering. 1D dynamic phantom sensations rendered by temporal inhibition [52] or amplitude inhibition [87] can provide clear sensations of directional movement on a mobile device. 2D dynamic phantom sensations have also been tested to find methods to render distinct 2D patterns [88, 105]. A recent study published IT of 3.70 bits using 2D illusory vibrotactile flows moving around the edges of a smartphone [88].

2.3 Automated Vibrotactile Feedback Authoring

Vibrotactile feedback has been used in interactive applications for a variety of purposes [20]. In such applications, use of the vibrotactile stimuli appropriate to their aims is of paramount importance. Designing such stimuli, however, is a challenging and time-consuming task and often exacerbated by the lack of suitable authoring software. Many researchers have been studied for vibration authoring to alleviate this problem by developing graphical authoring tools, automatic conversion of sounds to vibrations, automatic generation by using a physics engine, and mobile haptic rendering using a physics model. Below sections classify and briefly comment the vibrotactile feedback authoring methods.

2.3.1 Graphical Authoring Tools

Researchers have developed graphical authoring tool to facilitate the design of vibrotactile effects. Notable achievements include a vibrotactile waveform editor named Haptic Icon Prototyper [94]; a similar waveform editor with extended support for multiple actuators and a visualization of expected perceptual consequences [82]; a graphical editor using a metaphor to musical scores [57, 60]; a demonstration-based editor that converts a user's finger presses on a touchscreen to vibrotactile parameters [37]; and TactiPEd that supports multiple actuators using a graphical metaphor to the spatial layout of the device [73]. Although these editors improve productivity to great extent, the design process to obtain high-quality, well-tailored vibrotactile effects still depend on manual authoring, which can still be costly. Furthermore, such hand-made vibration patterns must be imported into application programs, and when and how to provide them to users also need to be programmed.

2.3.2 Conversion of Sounds to Vibrotactile Effects

An alternative approach is automatic generation of vibrotactile effects, preferably in real time. Sound has been the most popular source, initiated by an early work that sends an audio stream directly to a vibrotactile actuator while taking care of the bandwidth difference [17]. This method was extended by providing dual-channel vibrotactile playback in the bass and treble bands using a haptic equalizer [40], and further by making use of auditory saliency estimation [39]. A framework and algorithms for perception-level translation were also proposed [58], which enables selective audio-to-vibrotactile conversion by matching the perceptual characteristics such as roughness and loudness between sound and touch. In addition, research has recently begun to use images and videos as the source of conversion, e.g., by detecting visually salient areas on a video and emphasizing them with spatial tactile feedback [50].

2.3.3 Automatic Authoring Using a Physics Engine

It nevertheless remains unchallenged how to provide plausible vibrotactile feedback for various physical events in interactive applications with minimal developmental cost. A general solution for that is to use a physics engine, which simulates with high accuracy the dynamic behaviors of the objects involved in physical interaction. For instance, PhysX, a real-time physics engine of nVidia, was employed to provide realistic force feedback when collision events occur in a virtual environment [16]. To our knowledge, no such integrated efforts have been reported for vibrotactile feedback, despite its more frequent uses in interactive applications owing to greater affordability.

2.3.4 Physical Modeling of Sound

Another approach that might enable physically-accurate vibration synthesis is to adapt physics-based sound synthesis methods. The SOb European project classified prior research on sound synthesis to low-level physics-based models and higher-level structures [79]. In particular, various synthesis models of impact sound have been developed [4, 23, 27, 100]. Such methods were combined with a force-feedback system for multi-modal rendering [3, 27, 64, 91], where the user's contact force estimated from a force-feedback device is fed to the input of sound synthesis algorithms. Note that force feedback itself is done using the conventional constraint-based algorithms. Such sound synthesis algorithms are also generally inappropriate for vibrotactile feedback as they are. They are excessively complicated, resulting in a number of high-frequency modes and details that are imperceptible by touch.

2.3.5 Vibrotactile Rendering Using a Physics Model

Lastly, it is worth mentioning that a few previous studies employed a "shake the box" metaphor for non-visual interaction with a mobile device. A user shakes a mobile device (the box) to retrieve *abstract* information contained in it, and this action moves inner objects each representing a piece of information, e.g., the presence and property of a text message. If the objects collide with the box, predesigned sound or vibrotactile stimuli are presented to the user to deliver the meanings associated with the balls, similarly to tactons [10] and haptic icons [67]. This idea was first presented in [85], where vibrotactile cues are produced based on simple 1D dynamics simulation, but without auditory cues. This work was refined in Shoogle [104], which models the inner objects as balls and computes their motion using an anchored-ball dynamics model. Shoogle provides both auditory and vibrotactile stimuli. The latter used four sinusoidal waveforms with different envelopes, but they had the same frequency (250 Hz) and constant amplitudes regardless of the mass and the impact velocity of the ball, probably to facilitate the stimulus-meaning association. Both studies include simple dynamics simulation only for the collision detection between inner objects with simple geometry and the box, which is sufficient for their purpose. They do not (and need not) support a continuous modulation of vibrotactile stimuli in general virtual environments on the basis of accurate rigid-body dynamics simulation, which is the aim of our first application, PhysVib.

Chapter 3

Haptic Enchanter System: Localized Multi-Vibrotactile Feedback System

The aims of this work are twofold: (1) to propose haptic enchanters as a new conceptual vibrotactile feedback platform and 2) to address their fundamental advantages as an information presenting interface. For (1), several prototypes of haptic enchanters are designed to encompass vibration actuators of different form factors and demonstrate different attachment methods to rigid objects, human body, and wearables. These prototypes can be fabricated with or without a vibration attenuation layer, which enables vibration transmission to be isolated to a small region of contact. For (2), a user study is carried out to estimate the information transmission capacity in the absolute identification paradigm.

As referred above and in Chapter 1, haptic enchanters have two main requirements: 1) attachable to and detachable from user-desired body locations and 2) able to propagate vibrations to an attached object or to isolate vibrations generated from a haptic enchanter to a contacted site. By considering these, this chapter consists of two main sections: 1) development of a haptic enchanter system and 2) information capacity estimation of spatiotemporal patterns using haptic enchanters.

3.1 Structure of Haptic Enchanters

As a first step to develop such device, previous vibrotactile feedback systems are investigated to find out an appropriate stimulating body location that is sensitive enough to discriminate various vibrotactile stimuli, such as the Buzzwear on wrist [61], TVSS systems on back [6, 22, 103], or moving sensations on hands [49, 51, 87, 106]. Among them, the perceptual sensitivity of human hands is one of the most sensitive body locations and appropriate to implement an information system with high fidelity for its sensitivity to vibration intensity, frequency, and two-point discrimination [33]. Thus, haptic enchanters are designed to provide localized vibrotactile feedback to human hands using a mobile phone for the intra-hand interaction and non-localized vibrations to human fingers for the inter-hand interaction.

Previous studies provided localized vibrations by using three approaches: 1) directly attaching actuators to a human body part, 2) using a physically isolating structure, and 3) adding a vibration-isolating material between a contactor and a main device. First, human skin and flesh absorb vibrations well, therefore many researchers contacted actuators to human skin directly to provide spatial or spatiotemporal vibrotactile feedback. For example, a drum guidance system attached eccentric rotating mass (ERM) actuators to human torso and a leg for an education of playing a drum [56]. BuzzWear put three actuators around human wrist to provide spatiotemporal vibrotactile patterns [61], and Ubibraille let participants contact their fingers to six ERM actuators and represented alphabets using the actuators as a braille display [70]. Second, the vibrating contactor is physically separated from the main device to prevent the vibration propagation. SemFeel attached five actuators on a U-shaped case for closer attachment between human skin and the actuators, and this case attenuated vibration propagation for vibration localization [106]. Similarly, T-Mobile put 3 by 4 vibrating panels that mechanically attenuates vibrations on the back of a mobile phone, and a linear-resonance actuator was installed in each panel [105]. A Sony's dual-shock controller embedding two ERMs was designed to grab each stimulator module using each hand and a narrow middle-part connected the two modules, and this structure helps a user discriminate stimulated location individually. An approach adding an vibration isolating material has two good examples. T-Hive is a hemi-spherical device that added isolation material, polyurathane, between a stimulator and a main device to prevent vibration propagation [81]. HoliBraille estimated vibration attenuation performances using dampening materials of a spring, sorbothane, and cork, and a spring showed the highest attenuation than cork and sorbothane [71].

In this thesis, haptic enchanters are designed as a conceptual *platform*, not specific technologies. The ideal realization of haptic enchanters must include the design and development of individual technologies, e.g., actuators, electronics, and power, and their integration and packaging into one small module, all made adequate for the target applications. This would allow each haptic enchanter to be stand-alone and communicate individually with a control computer, providing ample extensibility of use. Attachment methods adequate for the applications also need to be devised.

Instead of developing such high-tech solutions, which would require years of significant engineering efforts, several prototypes of haptic enchanters are designed and fabricated to support commercial actuators to showcase the key concepts and features. These prototypes were used in user studies to examine the fundamental advantages of haptic enchanters as a communication means. This section presents the prototypes and discusses their potential applications.

3.1.1 Hardware Design

A haptic enchanter consists of a tactile actuator, an attenuation layer, and an attachment mechanism. For actuators, haptic enchanter supports three standard vibrotactile actuators at the moment: a coin-type ERM (Eccentric Rotating Mass) or LRA (Linear Resonant Actuator) widely used for smart phones and gaming pads, a cantilever-type piezoelectric actuator adopted in recent tablets, and a bar-type voice-coil actuator providing the best performance for vibration strength and frequency bandwidth. For the first two, the specific specifications of the models are: LRA—$\phi\,9\times3.4$ mm; resonance frequency 205 Hz and piezoelectric—$3.8\times3.2\times35$ mm; resonance frequency 230 Hz, both from Samsung Electromechanics. For

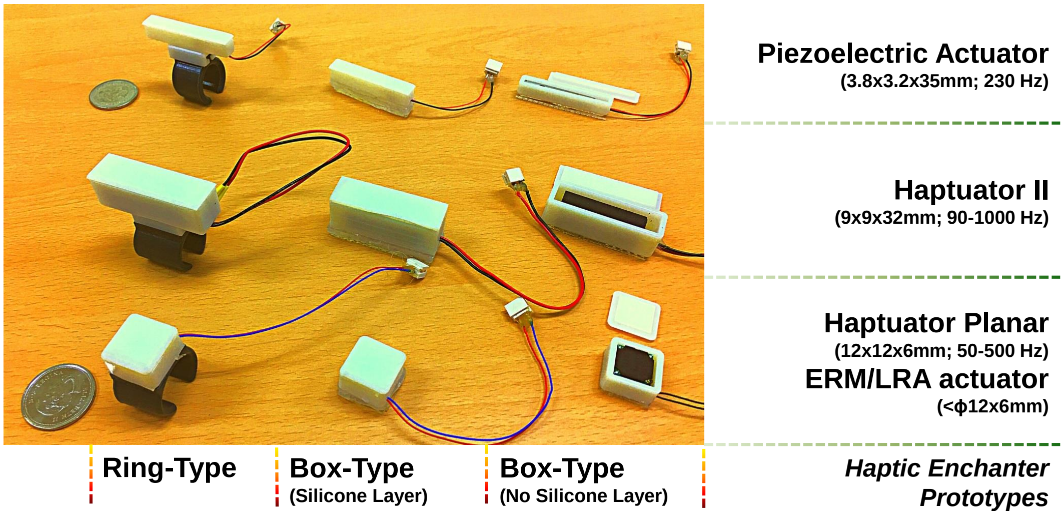


Fig. 3.1: Various prototypes of haptic enchanters. The case for Haptuator Planar can house most coin-type ERMs and LRAs used in mobile devices.

the last one, a haptic enchanter embeds a Haptuator (Mark II; 9×9×32 mm; frequency band 90–1000 Hz) from Tactile Labs. Three types of rigid case are made for the three actuators using 3D printing materials (acrylic-based photopolymer; VeroWhite RGD835 and VeroCyan RGD841).

An effective attachment/detachment mechanism depends on the shape of an object and its surface property. For devices with rigid and flat surfaces, e.g., smartphones and tablets, an anti-slip silicone pad (Spider Grip Pad; M-First) is selected after extensive testing of various methods such as an adhesive tape, an air sucker, and a high-frictional sticky material. The adhesive tape is not semipermanent although it is easy to attach to and detach from the rigid and flat surface, and the air sucker easily detaches from the surface when a shear force is given. Unlike the other materials, the anti-slip silicone pad allows easy and convenient attachment and detachment without any residue on the device surface while transmitting vibration energy without noticeable loss (Figure 3.1). For rigid wearables, e.g., rings, a T-shaped mechanical adaptor is available for rigid connection (Figure 3.1). This adaptor is plugged into a ring made using 3D printing (TangoBlackPlus FLX980; somewhat soft material for the ring's flexibility). All of these are merely some examples for a demonstration

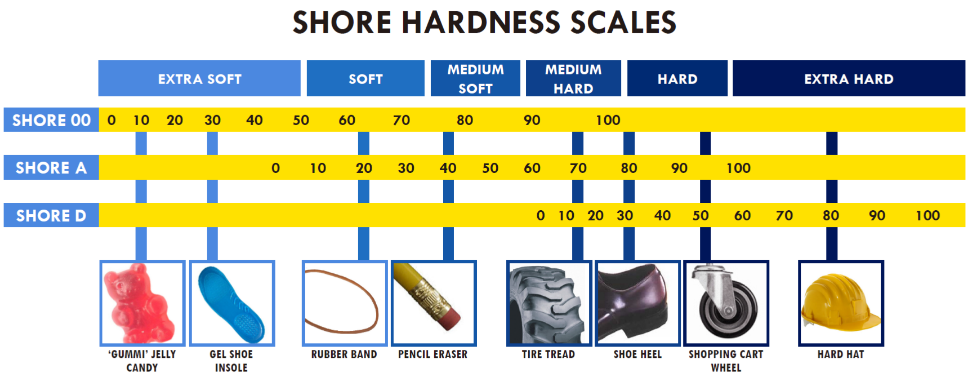


Fig. 3.2 Shore hardness scales for different materials. Shore 00: rubber and gel that are very soft, Shore A: flexible mold rubber and semi-rigid plastic, Shore D: hard rubber and hard plastic.

of the concept; one can always seek more sophisticated designs and attachment methods depending on the purpose.

For these prototypes, an external embedded board (NUCLEO-F334R8; Mbed) and two dual-DAC (digital-to-analog conversion) chips (TLC7528; Texas Instruments) are used to generate vibration signals to haptic enchanters. These signals are amplified by a custom circuit. The embedded board communicates with a computer or a smartphone via Bluetooth at 115,200 bps, driving up to four enchanters simultaneously.

Whether to have a vibration attenuation layer is optional for haptic enchanters. For instance, spatiotemporal feedback on a rigid device or object requires effective vibration isolation; otherwise vibration propagates and the entire object is shaken. Haptic enchanters for direct attachment to the user's body should not have a vibration attenuation mechanism so that their vibration is propagated to the user's skin with minimal energy loss.

To make a vibration attenuation layer, silicone mixtures are chosen since their viscoelastic properties are similar to the human skin and can be controlled by mixing additional chemicals. The extent of the hardness is generally measured by using shore hardness scale (Figure 3.2). To find the best silicone mixtures, a large number of silicone mixtures are made using two silicone mixtures of shore hardness 0010 and 0030 (EcoFlex Supersoft

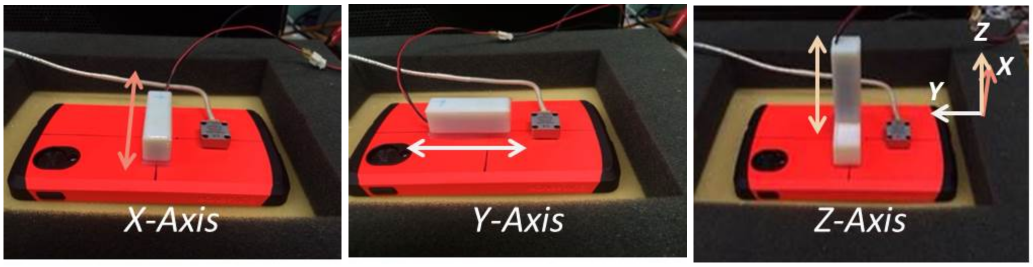


Fig. 3.3 Vibration measurement setup to find the best isolation material. A white box is a haptic enchanter enclosing a Haptuator.

0010 and 0030; SMOOTH-ON) as the base materials. These mixtures become softer when a softener (Slacker; SMOOTH-ON) is added. A large number of silicone mixtures are tested using 0010 silicone mixtures with softener ratios from 0 to 40% and 0030 silicone mixtures from 0 to 25%, both with step size 5%. The silicones mixtures had a thicknesses of 1.5, 3, 4.5, and 6 mm.

To measure vibration amplitudes with a wide frequency range (50–500 Hz), a haptic enchanter that included a Haptuator II on a smartphone (Nexus 5; Google; Figure 3.3). Since the Haptuator vibrates mainly along in the length direction, its attachment orientation was changed to measure vibration attenuation ratios in different directions. Acceleration data were measured using a triaxial accelerometer (8794A500; Kistler) located nearby the haptic enchanter. Measurement conditions differed in vibration frequency (50 to 500 Hz with 50 Hz steps), voltage (0.6 to 3.0 V with 0.6 V steps), and direction (x, y, and z; note the coordinate system in Figure 3.3), which resulted in 9,000 conditions.

Under each condition, acceleration values were measured from the vibrations transferred to the smartphone surface with the corresponding silicone layer between the haptic enchanter and the phone, for 0.4 s with 10 repetitions (90,000 measurements in total). As a reference, vibration acceleration values from the haptic enchanter itself was measured. By using the two acceleration values, a vibration attenuation ratio γ along the axis i was computed by

$$\gamma_i = 1 - \frac{A_i}{\hat{A}_i}, \tag{3.1}$$

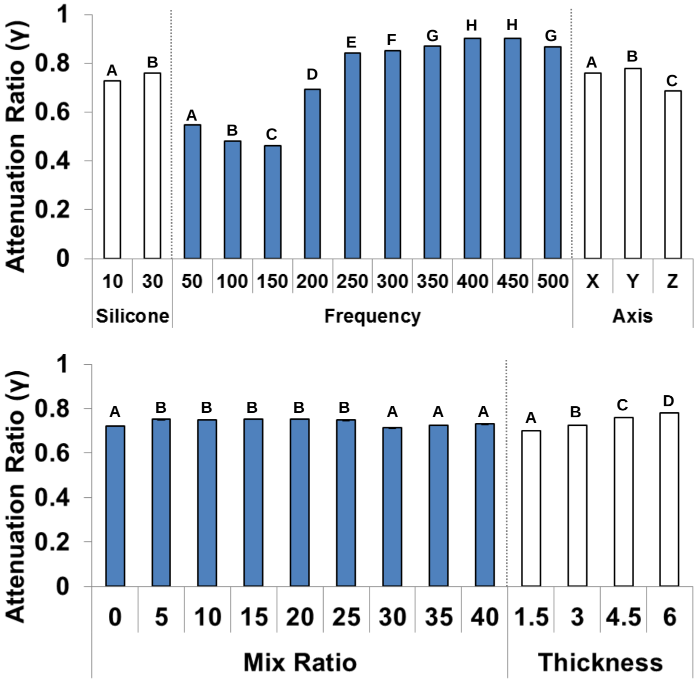


Fig. 3.4 Mean attenuation ratios for the main factors of silicone type, vibration frequency, excitation axis, mix ratio, and thicknesses. For each factor, conditions labeled with different alphabets had statistically significant differences.

where A_i is the attenuated (transferred) vibration amplitude and $\hat{A}_i I$ is the reference vibration amplitude. Then the attenuation ratio of the measurement condition was determined by

$$\gamma = \left(\gamma_x^2 + \gamma_y^2 + \gamma_z^2\right)^{\frac{1}{2}}. \tag{3.2}$$

This was necessary since vibrations in the other directions than the main excitation direction also showed non-negligible amplitudes due to the crosstalk.

The large number (9,000) of the experimental conditions does not allow to present individual attenuation ratios. Instead, a statistical analysis was carried out to examine the effects of main factors. Most of the measurement data did not follow a normal distribution (Shapiro-Wilk normality test). Hence, the Mann-Whitney's U Test were applied to the silicone type, and 0030 silicone mixtures had significantly larger attenuation ratios than 0010

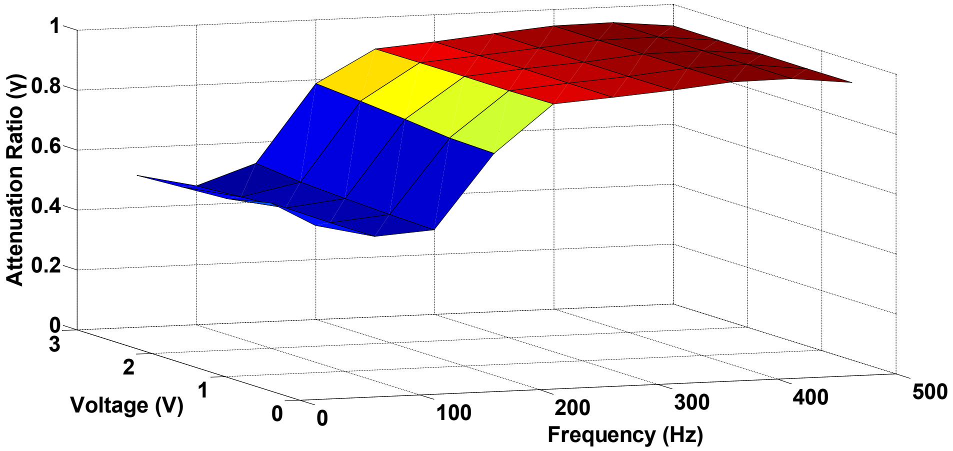


Fig. 3.5 Attenuation ratios (γ) of the silicone layer 0030–15 %–3 mm.

mixtures ($p < 0.01$). The other main factors were analyzed using the Kruskal-Wallis Test, and all of them were statistically significant for attenuation ratio ($p < 0.01$) except voltage. The means of the five significant main factors are shown in Figure 3.4, along with the results of post-hoc multiple comparison tests (Wilcoxson signed-rank test with Bonferroni corrections).

This analysis showed that the combinations of silicone 0030–(5–25)% mix ratio–6 mm thickness resulted in the highest mean γ values. However, 6 mm is too thick for small accessories to grab so this thickness implies a possibility of degraded usability of haptic enchanters. Instead, the mean γ values of the 0030–3 mm conditions were also very high, and a silicone mixture layer with 15% mix ratio showed $\gamma = 0.90$ over 200 Hz. A plot of γ for the 0030–15 %–3 mm silicone layer is provided in Figure 3.5 as a function of vibration frequency and applied voltage. It can be seen that this layer provides very effective vibration attenuation for a wide range of vibration frequencies, and its performance is independent of applied voltage. This silicone layer has been used in haptic enchanter prototypes, and also in user studies described in the following sections.

3.1.2 Applications

Haptic enchanters are a versatile platform that can enlarge the application areas of haptic feedback to the great extent. It is expected that haptic enchanters could improve the usability of existing applications and so increase the possibility of adoption in actual use. For example, there exists a good body of research as to how to re-create the feel of writing or drawing with realistic texture feedback on a touchscreen [80, 89]. However, such technology is not in actual use because of severe difficulties and cost increase of adding a wideband vibration actuator into a mobile device or a stylus. Attaching a haptic enchanter to a stylus and rendering textures using haptic augmented reality [45] on the basis of needs, e.g., when drawing an artistic painting, may prevent the glassy feel of the tablet's touchscreen from interfering with the user's imagination and concentration.

There have also been a plenty of research on a navigation aid for both sighted and visually-impaired users providing spatially-coded vibrotactile feedback, e.g., via a smart phone [44] or a belt [101]. Such technology, however, is not available in the market yet; to business decision makers, adding multiple actuators to a mobile device or a belt to help navigation may not seem to have pressing needs compared to the much increased cost. In this situation, haptic enchanters designed as a low-cost accessory of the main device or wearable may provide an alternative to the users in need of that function, e.g., for visually-impaired users to find an office in a big government building or for elderly drivers who need intuitive guidance of driving direction to the hands or fingers in a complex downtown instead of looking at the small GPS screen.

Haptic enchanters also implies a possibility to span design space for new applications. For instance, an interesting study in [96] verified that haptic feedback is effective in notifying timings during a technical presentation using a special wristband-type device. A haptic enchanter attached to a regular wireless presenter should provide the same function with the support of an appropriate smartphone app (Figure 3.6, top).

Besides, tactile feedback can enhance the user experiences of music listening [39, 40] and game playing [58], and it allows hearing-impaired users to feel musical beats and main melodies [69]. Using ring-type haptic enchanters, hearing-impaired users can experience

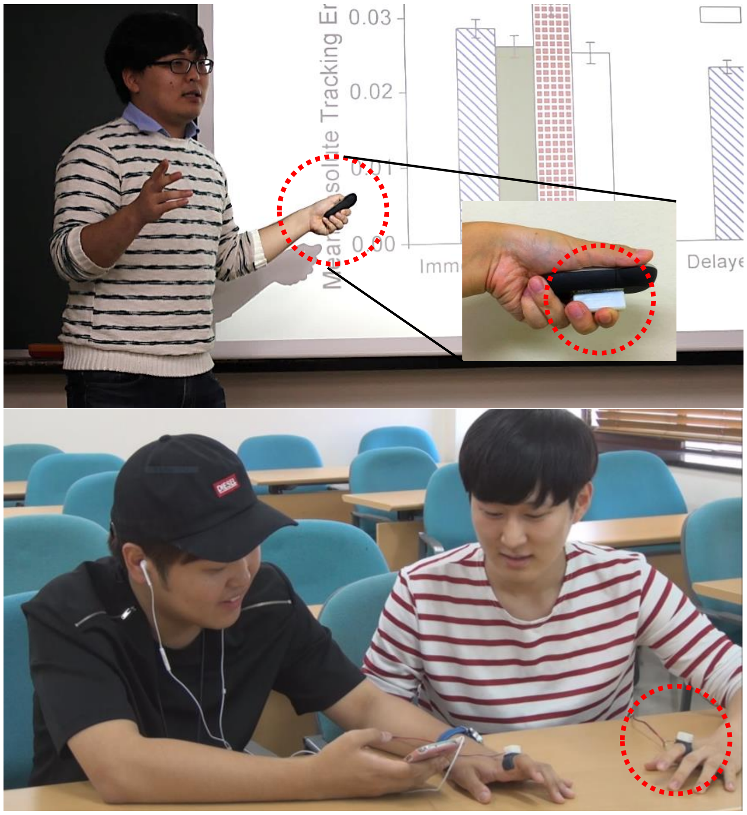


Fig. 3.6 Potential applications of haptic enchanters: presentation timing aid (top) and music sharing (bottom) using haptic enchanters. Objects enclosed by a red dotted circle provide vibrotactile feedback using haptic enchanters.

the musical beats and main melodies of music through vibrations while their friends with normal hearing ability listen to the music at the same place, enabling them to share the same activity (Figure 3.6, bottom).

The above user scenarios are by no means exhaustive. There can be many more creative, useful applications and use scenarios of haptic enchanters.

3.1.3 Advantages

Probably the most important advantage of haptic enchanters is that they extend regular devices and objects to haptic interfaces—the idea of adding haptics using stand-alone modules, not putting haptic components into the devices. This improves the possibility of many useful applications envisioned only in research papers to be realized in actual use.

The major benefits of haptic enchanters as a sensory display originate from the fact that they greatly facilitate use of multiple stimuli to provide spatial or spatiotemporal feedback. Stimulus isolation is essential for that; otherwise the stimulus energy propagates all over the device and the user cannot perceive spatial feedback. Vibration-attenuated haptic enchanters provide a very similar or equivalent situation to those in which multiple actuators directly stimulate different sites on the skin. Therefore, fundamentally, the benefits of haptic enchanters as a sensory display are determined by the human ability of processing spatiotemporal vibrotactile patterns. This is what the user studies in this thesis explain and next sections describe it: the information transmission capacity of the tactile sensory channel using vibrotactile patterns provided by haptic enchanters. User studies in this thesis were approved by the Institutional Review Board at the authors' institution (PIRB-2016-E020 and PIRB-2016-E041).

3.2 User Study: Information Capacity Using Spatiotemporal Pattern

User Study 1 pertains to the information transmission capacity of touch for spatiotemporal patterns enabled by the use of several haptic enchanters. The patterns were designed not to induce any illusory sensation. This is the most basic setup for information communication using haptic enchanters.

3.2.1 Methods

Two representative configurations of haptic enchanters are used as shown in Figure 3.7. In the PHONE configuration, haptic enchanters each enclosing an LRA with a vibration attenuation layer (0030–15 %–3 mm) were attached to the back panel of a smartphone (Nexus 5; Google). Participants grasped the smartphone using their left hand and perceived localized vibration patterns. In the RING configuration, participants wore ring-type haptic enchanters each including an LRA without a vibration attenuation layer in their fingers. The former represents the most common use case of attaching haptic enchanters to a device

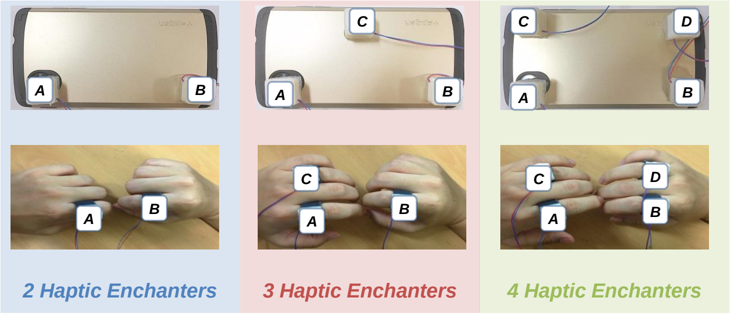


Fig. 3.7 Haptic enchanters attached to a smartphone (top; PHONE) and worn in the fingers as rings (bottom; RING). In User Study 1, two, three, or four haptic enchanters were used for each configuration. User Study 2 used only the four haptic enchanter setups.

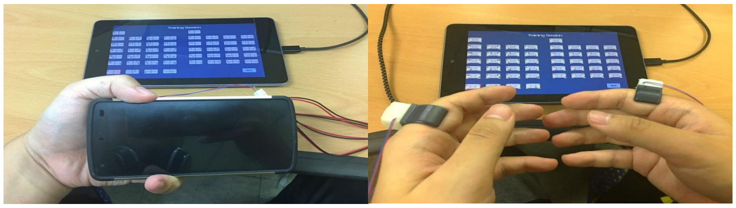


Fig. 3.8 Hand postures used in both user studies. (Left: PHONE) Participants grasped the phone using their left hand. (Right: RING) Participants held their two hands as if they were holding a phone in the landscape mode. Participants used the tablet placed on the table to enter commands and responses during the experiment.

that users carry everyday, while the latter is an example of convenient direct skin stimulations via wearables. Each enchanter was marked with an alphabet (A–D) for participants' easy recognition. Hand postures used in both user studies are shown in Figure 3.8.

For each enchanter configuration, vibration patterns were rendered using two, three, or four haptic enchanters at different locations (Figure 3.7). Changing body site is simple but very effective for stimulus identification [12, 20]. Each vibration pattern consisted of one, two, or three short pulses, and each pulse was triggered with one of the enchanters. Examples of the vibration patterns are shown below:

- 2 enchanters: A, B, A→A, B→B, A→A→B, B→B→A
- 3 enchanters: A, C, A→C, B→C, A→B→C, B→B→C

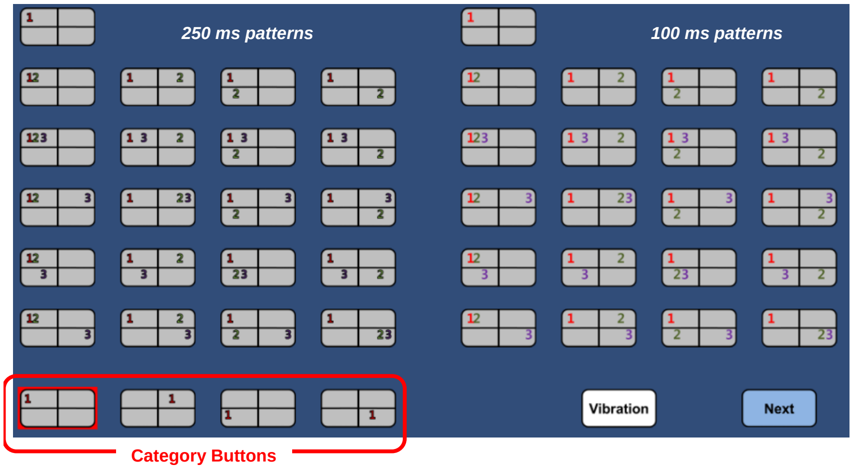


Fig. 3.9 GUI of the experiment program used in User Study 1. Participants presses one of the four category buttons for the location of the first pulse. Then the program displays icons for all possible patterns beginning with the first pulse location entered. The left icon group is for 250-ms patterns, while the right group is for 100-ms patterns. Participants then choose the corresponding icon for the final answer. The design of the icons is illustrated further in Figure 3.10. This procedure allows participants to enter a response with only two clicks in spite of the very large number of alternatives.

- 4 enchanters: B, D, D→A, D→B, A→D→C, D→D→C

Using a greater number of pulses is prone to a higher error rate because of the underestimation bias in the human temporal numerosity judgment [55, 77]. All pulses were sinusoidal vibrations with the same amplitude of 1 g[1] and frequency of 205 Hz (the resonance frequency of the LRA used). The inter-pulse interval was 100 ms. Lastly, the duration of each vibration pulse was 100 or 250 ms, and the same duration was used for all the pulses in one pattern. The two values were selected to be sufficiently long and maximize their discriminability [99], while maintaining the entire patterns to be shorter than 1 s. The total number of vibration patterns were 28, 78, and 168 for 2, 3, and 4 haptic enchanters.[2]

Participants' task was that of absolute identification; participants perceived one vibration

[1] Note that this italic g is a standard notation for the gravity acceleration constant. It is not g for gram (a unit of weight).

[2] The number of patterns $= D(E + E^2 + E^3)$ where D is the number of durations and E is the number of haptic enchanters.

pattern and answered which pattern it was based on absolute judgments. Participants used another tablet (Nexus 7; Google) running the program shown in Figure 3.9 with a graphical user interface (GUI). On each trial in the main sessions, participants first clicked the 'Vibration' button to feel the vibration pattern and then entered the identity of the pattern using the GUI. The GUI was designed to allow participants to choose answers with only two button clicks in spite of the very large numbers of patterns.[3] The first click was for the location of the first pulse, and it was made on one of the four category icons located in the bottom left of the screen. Then the program displayed icons for all possible patterns beginning with the first location entered. Figure 3.9 shows an example after a participant pressed the leftmost button among the four category icons in a session where four haptic enchanters were used. The main screen presents two groups of icons for different pulse durations. The design of each icon is illustrated in Figure 3.10. Then participants selected the corresponding icon, and this second click finished the trial. Participants were allowed to perceive vibration patterns repeatedly by clicking the 'Vibration' button again when necessary. To proceed to the next trial, participants clicked the 'Next' button. Participants wore headphones that played pink noise to block the weak sound produced by the LRAs.

Participants completed the experiment in two days: one day for PHONE and the other day for RING. The experiment for each configuration consisted of three sessions using two, three, and four haptic enchanters, respectively. Participants finished one training session and one main session for each combination of the two configurations and the three number of haptic enchanters. The order of configuration and the number of haptic enchanters was balanced using Latin Squares.

The training session was to allow participants to experience all vibration patterns at least once and become familiar with the interface shown in Figure 3.9. The 'Vibration' button

[3] An experiment to measure the IT of a sensory channel is designed in such a way that all memory-related factors, e.g., the need to remember stimulus-response codes, are removed. This is to isolate the channel capacity as a communication channel, not affected by the memory capacity, which is usually more limited. Thus, a table that depicts the stimulus-response mapping in an appropriate way is generally given to participants. All the related studies about the IT of tactile stimuli described in Related Work earlier did the same. In User Study 1, the number of stimuli was exceptionally large, and a means to make the answer selection efficient was necessary.

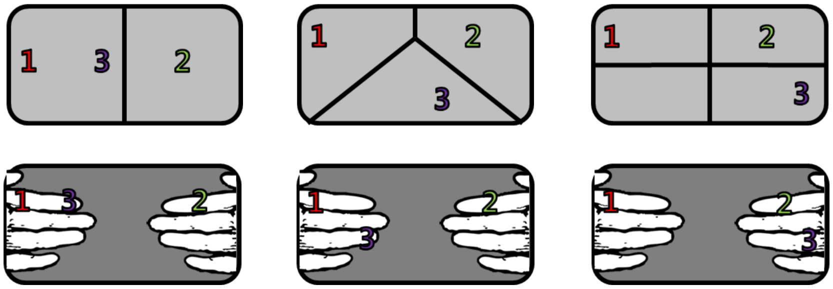


Fig. 3.10 Design of the icons used in the experiment program. From left to right, the icons in the top row represent A-B-A (two enchanters), A-B-C (three enchanters), and A-B-D (four enchanters) for PHONE. The order of vibration pulses was denoted by a color-coded number (1–red, 2–green, and 3–purple) within the icon and also by the relative position of the number (e.g., for A→A→A, 1 2 3 is displayed without overlap). The bottom row shows example icons displayed for RING.

was disabled, and participants had to select an icon button to perceive the corresponding vibration. The training session was finished after participants selected all vibration patterns. In the main sessions, vibration patterns were presented to participants in random order. Each vibration pattern was presented only once. This was necessary to maintain the size of the experiment manageable because of the very large number of the vibration patterns used. Participants took 2 minutes of rest after finishing each main session. They were also allowed to take a break whenever necessary. The experiment took approximately three hours for each participant to complete (about one and a half hours per day).

The experimental data collected in the main sessions were processed to estimate IT following the standard procedure [84, 98]. The data of all participants were pooled for each combination of the two configurations (PHONE and RING) and the three numbers (2, 3, and 4) of haptic enchanters and six stimulus-response confusion matrices were made. Then the maximum likelihood estimate of IT was computed using the standard formula in [98] from each confusion matrix. Percent correct (PC) scores were also calculated.

For the user study, 12 users (11 males and 1 female; 18–30 years old with a mean 24.6; all right-handed) were participated. None of them reported known sensorimotor impairments. They were paid approximately USD 30 for their voluntary help after the experiment.

3.2.2 Results

The IT values estimated in User Study 1 are shown in Figure 3.11. When two haptic enchanters were used, the IT values were both 4.55 bits for PHONE and RING. The IT values were increased to 6.03 and 6.06 bits with three enchanters, and further to 7.06 and 6.92 bits with four enchanters. The IT estimates of PHONE and RING were very similar.[4]

All IT values were very high in reference to the maximum IT achievable (4.81, 6.29, and 7.39 bits); all more than 92% of the maximum IT. These results instantiate that haptic enchanters can transform everyday devices and wearables to effective information displays in terms of the channel capacity.

The mean PC scores are shown in Figure 3.12. The scores ranging from 89.2% to 95.0% with the grand mean of 92.1%. These scores are considered as very high, especially for the very large number of the vibration patterns. Again PHONE and RING exhibited very similar PC scores for the same numbers of haptic enchanters. The PC scores tended to decrease with the number of haptic enchanters, but that was expected. Two-way ANOVA with repeated measures showed that configuration and the number of haptic enchanters did not have significant effects on PC score ($F(1, 11) = 0.21, p = 0.658$; $F(1, 11) = 4.62, p = 0.055$), albeit the p-value of the number of haptic enchanters close to the significance level ($\alpha = 0.05$).

3.2.3 Discussion

Comparisons with Previous Work

The IT estimates obtained in User Study 1 ranged from 4.55 bits (both PHONE and RING; two haptic enchanters) to 7.06 bits (PHONE; four enchanters). The highest IT of 7.06 bits means that 133 vibration patterns[5] can be identified without errors. This is the greatest among those reported in the literature for tactile information communication; see the review in Related Work.

[4] Only one IT estimate was obtained from a confusion matrix that pooled the data of all participants for each combination. Hence, statistical tests could not be performed.

[5] Given the sensory channel and the stimulus set, the number of stimuli that can be identified perfectly through the channel is 2^{IT}.

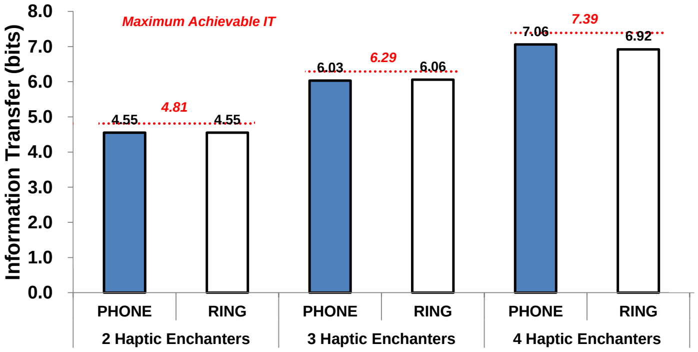


Fig. 3.11 IT estimates of User Study 1.

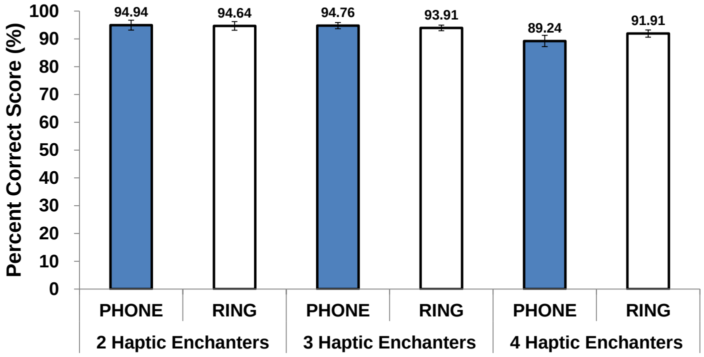


Fig. 3.12 Percent correct scores measured in User Study 1.

The vibrotactile patterns used were quite simple. Each pattern consisted of one to three short vibrotactile pulses applied at different locations. All pulses had the same frequency, amplitude, and inter-pulse interval. The only temporal factor was using two values (100 and 250 ms) for the pulse duration. Further, all patterns were short; the longest pattern was 950 ms long. Therefore, the design emphasized the spatial aspect of vibrotactile stimulation, and this was the enabling factor of such high information communication capacity.

This aspect can be better understood by comparing this study with the two related studies [97, 61] that presented the highest IT values prior to this work. In [97], a sophisticated multi-finger tactile communication device named Tactuator was developed. Tactuator was inspired by Tadoma (a speech recognition method used by deaf-blind individuals), and it can stimulate the thumb, index finger, and middle finger independently. This study used 120 stimuli delivered to the three fingers through 30 waveforms of different frequencies and amplitudes. The estimated IT was 6.50 bits corresponding to 91 patterns of absolute identification. BuzzWear in [61] used a wrist-band equipped with three vibration motors. The authors designed 24 vibration patterns by varying intensity, rhythm, starting point, and rotating direction. This work resulted in the IT of 4.28 bits (19 patterns).

Both of these two previous studies used three body sites for vibrotactile stimulation. The estimated IT with three haptic enchanters were 6.03 bits (PHONE) and 6.06 bits (RING), which were slightly lower than the IT of Tactuator but fairly higher than that of BuzzWear. The estimated IT with four enchanters were much larger (7.06 and 6.92 bits; 133 and 121 items with perfect identification). These comparisons, also considering the simpler design of vibration patterns presented here, suggest that using additional body site was the key in the improvement of IT with four enchanters. Also this was enabled by the modular nature of haptic enchanters; they can be easily attached to the objects or body sites that offer the best potential for effective identification.

Implications to Applications

This user study provided the specific values of IT for six configurations of haptic enchanters, and IT gives the maximum number of stimuli that can be identified without errors. Based on these findings, interaction designers can design tactile icons, brief vibrotactile patterns delivering abstract meanings [10], as follows [18]: 1) Given an application, decide how many tactile icons are necessary; 2) Choose the number of haptic enchanters that provides a sufficient IT for the number of tactile icons to be used; and 3) Select vibration patterns with the highest PC scores and also intuitive associations with the intended meanings. These results provide detailed foundations to all the three steps, leaving only the application-dependent meaning mapping step to application designers.

In general, the most challenging step in tactile icon design is to find the vibrotactile patterns that naturally remind of designated meanings [41]. This is not easy since people do not experience vibrations from most physical objects around them. Hence designers generally rely on metaphors for stimulus-meaning association [66]. In this regard, haptic enchanters have a merit since their spatial configuration can be easily adapted.

3.3 Conclusions

This chapter has described the design and implementation of haptic enchanters and its performance as an information channel without using illusory effect. Haptic enchanters consist of an attachment part, a vibration actuator, and a vibration attenuation layer (optional). Nine prototypes are developed for perceptual experiments to representative two configurations of PHONE and RING using haptic enchanters. Using these prototypes, a user can attach box-type haptic enchanters to a mobile phone and get localized vibrations and can feel localized vibrations by wearing ring-type enchanters on their fingers.

A user study reported the high performance as an information transmission system using a simple spatiotemporal coding without using any illusory effect. This coding is quite simple and requires short time how to learn the patterns. Even with these small efforts, using 2 and 3 haptic enchanters showed comparable estimated IT (4.55-6.06) and using 4 haptic enchanters provided the highest IT values (6.92 and 7.06) in the literature.

Chapter 4

2D Stationary Phantom Sensations Using Four Haptic Enchanters

In the previous chapter, an information capacity of haptic enchanters using spatiotemporal patterns without illusory feedback was investigated. The process is as usual as other literature and straightforward to admit the result.

In tactile sensations, phantom sensations have been investigated to provide spatial information with higher resolution by using a small number of actuators as referred in Section . In the literature, localized vibrotactile feedback provides more clear illusory perception of midway points between stimulated points, and PHONE perfectly fits to this case. However, researches of 'out-of-body' phantom sensations that can be rendered in RING are rare and there exists only one paper of its 2D case.

The perceptual performances of 2D phantom sensations were measured using the information transfer (IT) and the perceptual resolution in this chapter. The former is a novel approach to measure illusory feedback performance in the literature, and the latter is a measure used in the previous work of Kim's group [53].

4.1 User Study 1: Information Transfer Using 2D Phantom Sensations

User Study in Chapter 3 used the vibrotactile patterns consisting of a sequence of single pulses applied to different body sites. If multiple body locations are stimulated *simultaneously* and properly, it elicits a single illusory stationary sensation *within* the area enclosed by the stimulation positions. This phenomenon of phantom sensations is another powerful way of utilizing haptic enchanters. The aim of User Study 1 was to probe the fundamental information transmission capacity of 2D phantom sensations. This can enlarge our scientific understanding of phantom sensations since related studies to 2D stationary phantom sensations are rare as reviewed earlier.

4.1.1 Methods

This study used the same hardware of User Study in Chapter 3, but only the four haptic enchanter cases were tested for both configurations of PHONE and RING (Figure 3.7). PHONE is very similar to the standard setup of 2D phantom sensations, which attaches actuators directly to the skin, since the haptic enchanters included the vibration attenuation layer. RING tests out-of-the-body 2D phantom sensations [59]; illusory sensations occur in the air between the two hands.

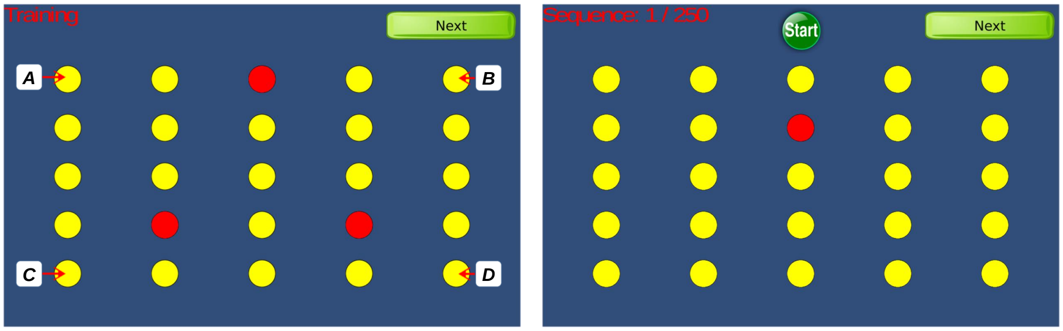
Let four haptic enchanters be located at the four positions A, B, C, and D. The target location of phantom sensation is within the rectangle ABCD. For that, the haptic enchanter i renders a vibration signal $x_i(t)$:

$$x_i(t) = A_i \sin(2\pi F t) \tag{4.1}$$

where $F = 205\,\text{Hz}$ for all i. The vibration amplitude A_i is computed by

$$A_i = A\left(1 - \frac{d_i^x}{D^x}\right)^{\gamma}\left(1 - \frac{d_i^y}{D^y}\right)^{\gamma}, \tag{4.2}$$

where A is the maximum amplitude (1 g), d_i^x and d_i^y are the horizontal and vertical distances from the haptic enchanter i to the target location, D^x and D^y are the maximum distances, and γ is a control parameter for the perceived quality of phantom sensation. This rendering



(a) (b)

Fig. 4.1 Screenshots of the application used in User Study 1. (a) training session and (b) main session.

rule is extended from the previous work on 1D phantom sensations [1, 87]. The duration of vibrations was 1 s. Various default values for γ were tested and then $\gamma = 1$ was chosen since it elicited the clearest location perception.

An application was designed for this user study and displayed 3×3, 4×4, and 5×5 grids on a tablet. An example of the 5x5 grid is shown in Figure 4.1. The 3×3 and 4×4 grids shared the same corner positions with the 5x5 grid. When the yellow buttons are touched, then 2D phantom sensations were provided using the rendering method in Eq. (4.2).

The experiment had six experimental conditions combining two enchanter configurations (PHONE and RING) and three virtual grids (3×3, 4×4, and 5×5). The order of the experimental conditions was balanced using Latin squares.

Participants finished one training session and one main session for each experimental condition. In the training session, participants touched a yellow circle on the tablet to feel the corresponding phantom sensation, and then the color of the circle turned red (Figure 4.1(a)). After experiencing all positions, participants clicked the 'Next' button to proceed to the main session, or they could repeat the training until they thought they were prepared. Note that clear perception of phantom sensations generally requires short-term training [1, 88]. The main session started after 2 minutes of break. On each trial, participants touched the 'Start' button (Figure 4.1(b)) to perceive a phantom sensation in random order and then clicked the yellow circle for the perceived location. Then participants pressed the 'Next' button to move to the next trial. The 2D phantom sensation stimulus for each grid point was repeated ten times. Hence, the total numbers of trials were 90 (3×3), 160 (4×4), and 250 (5×5).

Participants had 2 minutes of break before starting the session for the next experimental condition. Participants wore headphones that played pink noise to block the faint sound produced by the haptic enchanters.

The data collected in the main sessions were processed to estimate IT and PC scores as in User Study 1. This time two criteria were used for correct identification given the illusory nature of phantom sensations. One criterion required exact identification: if the stimulus is for (x^*, y^*), only the response (x^*, y^*) is regarded as correct. The other allowed one grid error: a response (x, y) is considered as correct if the Manhattan distance between (x, y) and (x^*, y^*) is no more than 1.

Eighteen participants (3 female and 15 male; 18–29 years old with a mean 22.7) were recruited in this study. No participants reported known sensorimotor disorders, and all participants were right-handed. The experiment took two hours on average for each participant. Participants were paid about USD 20 for their help after the experiment.

4.1.2 Results

Figure 4.2 shows the estimated IT values for each experimental condition. The maximum IT achievable with the 3×3, 4×4, and 5×5 virtual grids were 3.17, 4.00, and 4.64 bits, respectively. The IT estimates computed with the exact identification criterion without tolerance were 1.89, 1.77, and 1.76 bits for PHONE and 2.53, 2.32, and 2.28 bits for RING. These IT estimates decreased as more complex virtual grids were used. The IT estimates computed with 1-grid error tolerance were 2.76, 3.04, and 3.00 bits for PHONE and 3.05, 3.50, and 2.62 bits for RING. In this case, the highest IT values were obtained with the 4×4 grid. For both correct identification criteria, RING resulted in better channel capacity than PHONE.

The mean PC scores are shown in Figure 4.3. The mean PC scores were 74.0, 45.3, and 30.6% (PHONE) and 88.7, 61.7, and 45.1% (RING) without error tolerance. They

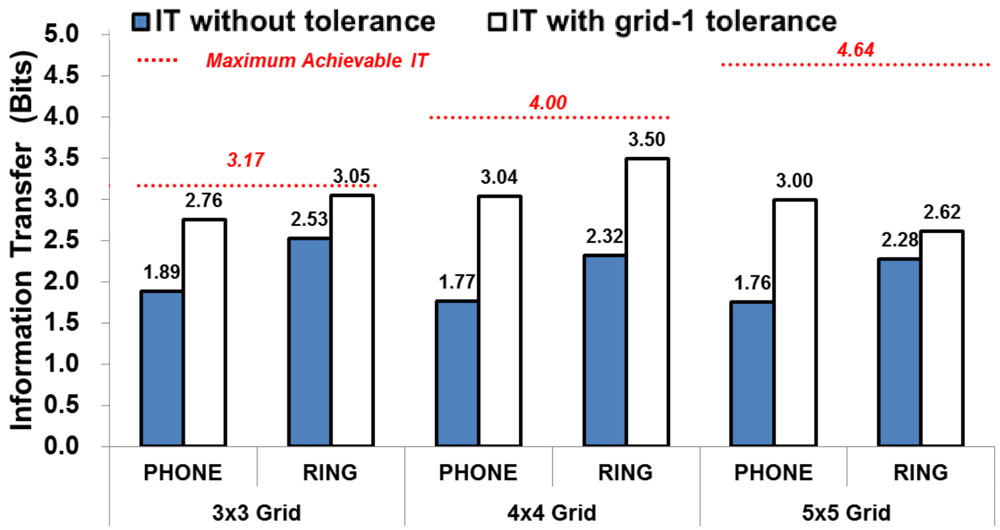


Fig. 4.2 IT estimates obtained in User Study 1.

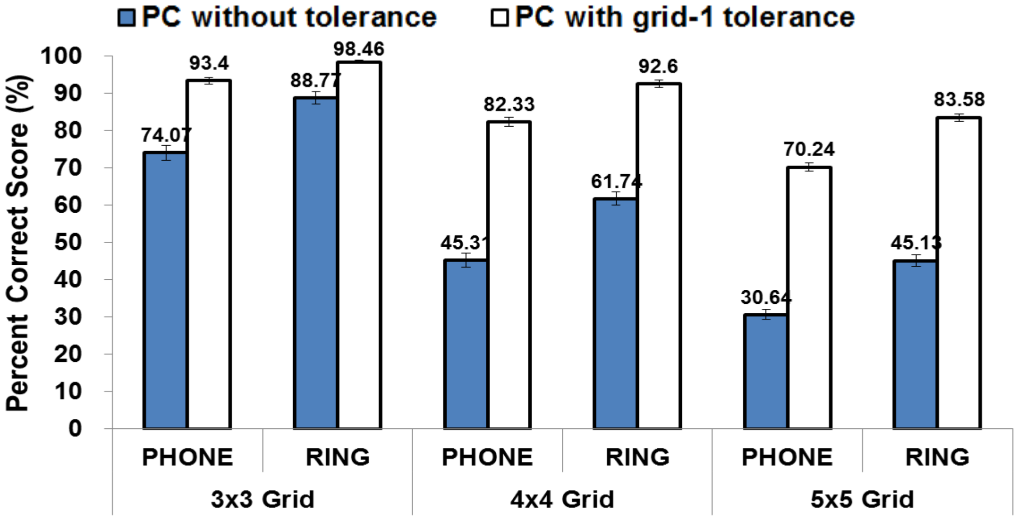
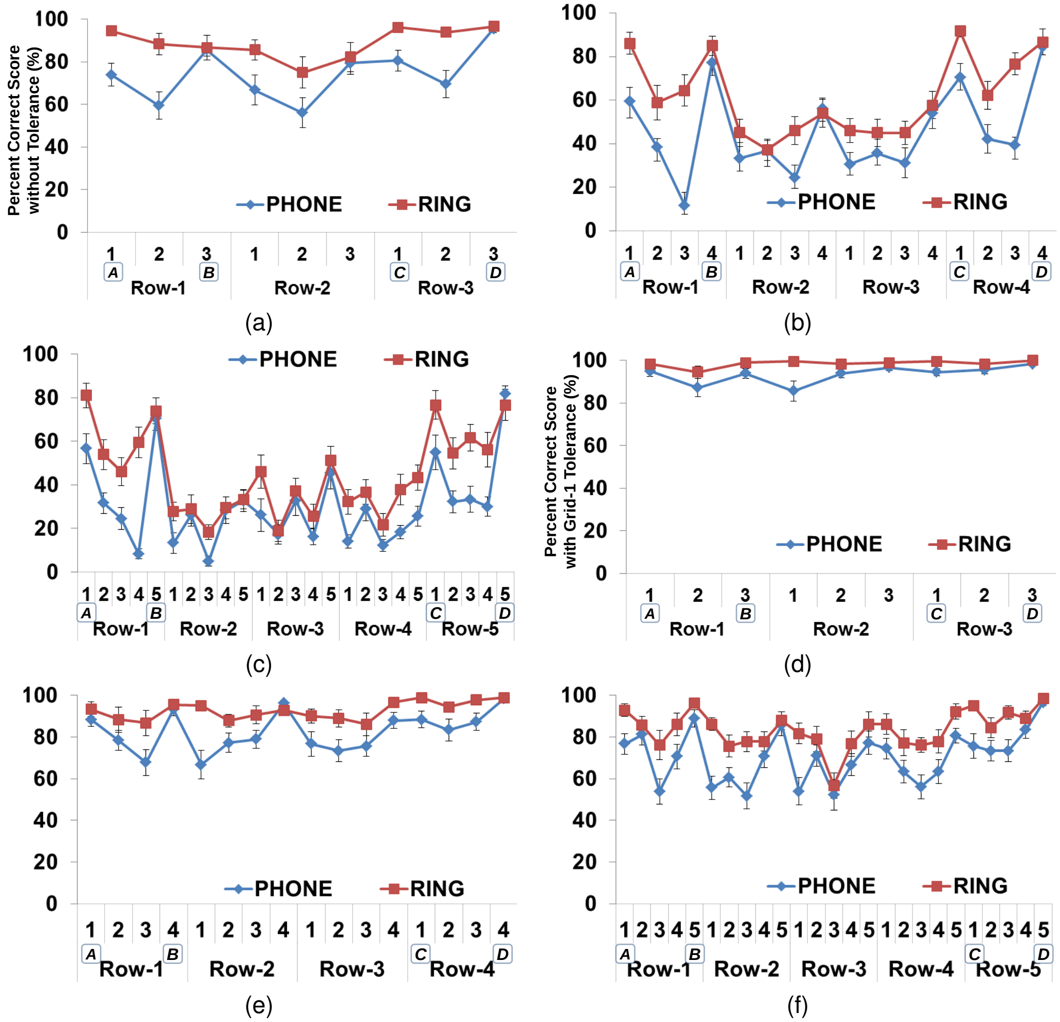


Fig. 4.3 Mean PC scores measured in User Study 1. Error bars show standard errors.

were 93.4, 82.3, and 70.2% (PHONE) and 98.4, 92.6, and 83.5% (RING) with 1-grid error tolerance. Both enchanter configuration and grid size had significant effects on PC score with and without error tolerance (Two-way ANOVA with repeated measures; $F(1,17) =$



(a) (b) (c) (d) (e) (f)

Fig. 4.4: Mean PC scores of 2D phantom sensations. (a, b, c): computed with no error tolerance and (d, e, f): computed with 1-grid error tolerance. (a, d): 3×3 grid, (b, e): 4×4 grid, and (c, f): 5×5. The numbers with alphabets (A–D) represent the grid points at which real actuators were located through haptic enchanters. Error bars represent standard errors.

$53.94, p < 0.001$ and $F(1,17) = 39.51, p < 0.001$; $F(2,34) = 573.4, p < 0.001$ and $F(2,34) = 69.35, p < 0.001$). Therefore, it is able to conclude that RING enabled better PC scores than PHONE, and the PC scores decreased with more alternatives in the grid.

To garner better insights, the PC scores of individual grid locations are shown in Figure 4.4. The plots commonly show that the PC scores were the highest at the grid locations where real actuators were placed through haptic enchanters and the PC scores were the lowest in the middle of the rectangles formed by the four enchanters; see (c) and (f) for the 5×5 grid. Other major observations were consistent with those made earlier using the mean PC scores.

4.1.3 Discussion

Comparison with the IT of Tactile Grid Display

To our knowledge, there exist no studies that estimated the IT of stationary 2D phantom sensations. The results of User Study 1 provide new scientific knowledge to the literature.

In [90], the authors estimated the IT of a 3×3 tactile grid display (9 vibration actuators; 30 mm gap between the actuators) applied to the palm. The vibration patterns used were sinusoidal with amplitude 7 *g* (very strong for clear perception), frequency 70 Hz, and duration 2 s. The estimated IT was 2.46 bits with 9 locations. This value obtained with real stimuli is higher than the IT values (1.76–1.89 bits) measured in the study under the PHONE configuration with the exact identification criterion. This comparison clearly demonstrates the nature of trade-off between using real stimuli with many actuators and illusory stimuli using several actuators.

Within vs. Out-of-the-Body Phantom Sensations

In the PHONE configuration, four haptic enchanters stimulate the palm and fingers to elicit phantom sensations within the skin area enclosed by the actuators. This "within" phantom sensation is what most researchers have studied [1, 43, 95, 105, 106]. In contrast, phantom sensations occur in the empty space between the two hands in RING. To our knowledge, this is the first report of this type of "out-of-the-body" 2D phantom sensation. Although similar out-of-the-body 2D phantom sensations were reported recently in [53], users held a large tablet with both hands and stationary phantom sensations were reported to occur within the tablet. There has been no known report of out-of-the-body phantom sensations when no rigid medium is present between the hands.

One very interesting finding of User Study 1 is that out-of-the body phantom sensations outperformed within-hand phantom sensations in both of the IT estimates and the PC scores (the latter even had clear statistical significance). It was initially expected that the opposite or similar performance would be estimated between them. No similar comparison results could be found in the literature. The best conjecture for the underlying reasons is that in PHONE multiple finger joints exist between the stimulation points on the palm and fingers and this might have weakened phantom sensations. Most prior studies did not have joints among the stimulation points [1, 7]. This topic deserves further attention.

Implications to Applications

The information transmission performance through stationary 2D phantom sensations was not sufficient to use them as a primary information channel. IT estimates under the exact identification criterion ranged from 1.76 to 2.53 bits, and the PC scores varied between 30.6% and 88.7%. These results are consistent with those of [53], which also showed low accuracies using out-of-the-body 2D phantom sensations. However, the IT and recognition accuracy of 2D dynamic phantom sensations reported in the literature were considerably higher. For example, edge flows reported 3.70 bits of IT [88], T-Mobile showed 92.2% of recognition accuracy using 10 patterns [105], and SemFeel accomplished 89.6% of accuracy with 11 patterns [106]. Therefore, dynamic 2D phantom sensations are recommended for interaction requiring the exact delivery of information, e.g., tactile icons. Note that most of the patterns for dynamic 2D phantom sensations mentioned above can be easily replicated using haptic enchanters.

The IT estimates and PC scores were substantially higher when the grid-1 tolerance error criterion was used (IT: 2.62–3.50 bits; PC score: 70.2%–98.5%). This result indicates that 2D stationary phantom sensations still occur around the target location, although the perceptual accuracy is not sufficiently high. Hence, 2D stationary phantom sensations could be better for secondary or ambient cues guiding the user's attention to the neighborhood of an event of interest in multimodal applications, e.g., for games and entertainment.

4.2 User Study 2: Spatial Resolution of 2D Phantom Sensations

User Study 1 showed that 2D stationary phantom sensations are not appropriate to provide primary information for their low IT and they could be better for secondary or ambient cues guiding the user's attention. However, the perceptual resolution of the 2D stationary phantom sensations is still unveiled and it is impossible to utilize 2D phantom sensations as a feedback of a spatial vibrotactile information system. Thus, the aim of User Study 2 was to probe the perceptual resolution of 2D phantom sensations.

4.2.1 Methods

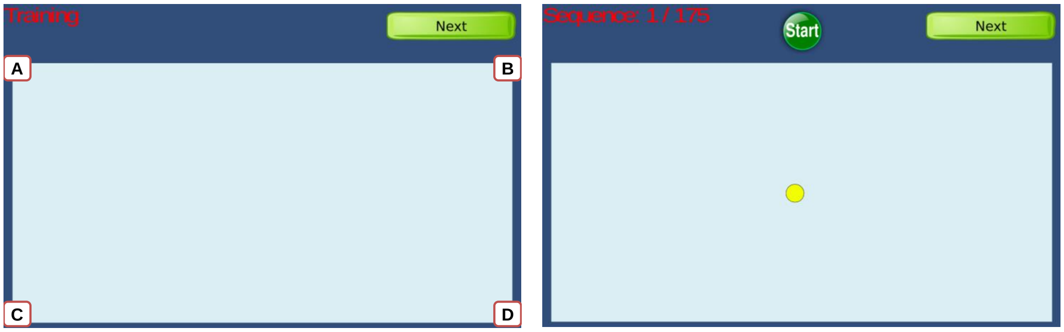
This study used exactly the same hardware of User Study 1 (Figure 3.7), and only one difference is that this study used two rendering methods for 2D phantom sensations: a linear rendering and a logarithmic rendering. The linear rendering calculates the amplitude of each actuator by using the equation 4.2 with $\gamma = 1$ and the logarithmic rendering uses

$$A_i = A\left(1 - log_2\left(1 - \frac{d_i^x}{D^x}\right)\right)\left(1 - log_2\left(1 - \frac{d_i^y}{D^y}\right)\right), \tag{4.3}$$

where A is the maximum amplitude ($1\,g$), d_i^x and d_i^y are the horizontal and vertical distances from the haptic enchanter i to the target location, and D^x and D^y are the maximum distances. This rendering rule is extended from the previous work on 2D phantom sensations [53] that is a unique reference of an out-of-body 2D phantom sensation..

An application was designed for this user study and displayed a gray square on a tablet as in Figure 4.5. Each corner was assigned to each actuator, and the midway points on the gray square were rendered using 2D phantom sensations with equation 4.2 or 4.3.

The experiment had four experimental conditions combining two enchanter configurations (PHONE and RING) same as in User Study 1 and two rendering methods (Lin–Linear and Log–Logarithmic). The order of the experimental conditions was balanced using Latin squares. The gray square was divided into a 7 by 5 grid and 35 phantom sensations were prepared and repeated 5 times, therefore a user responded 135 phantom sensation for each condition.



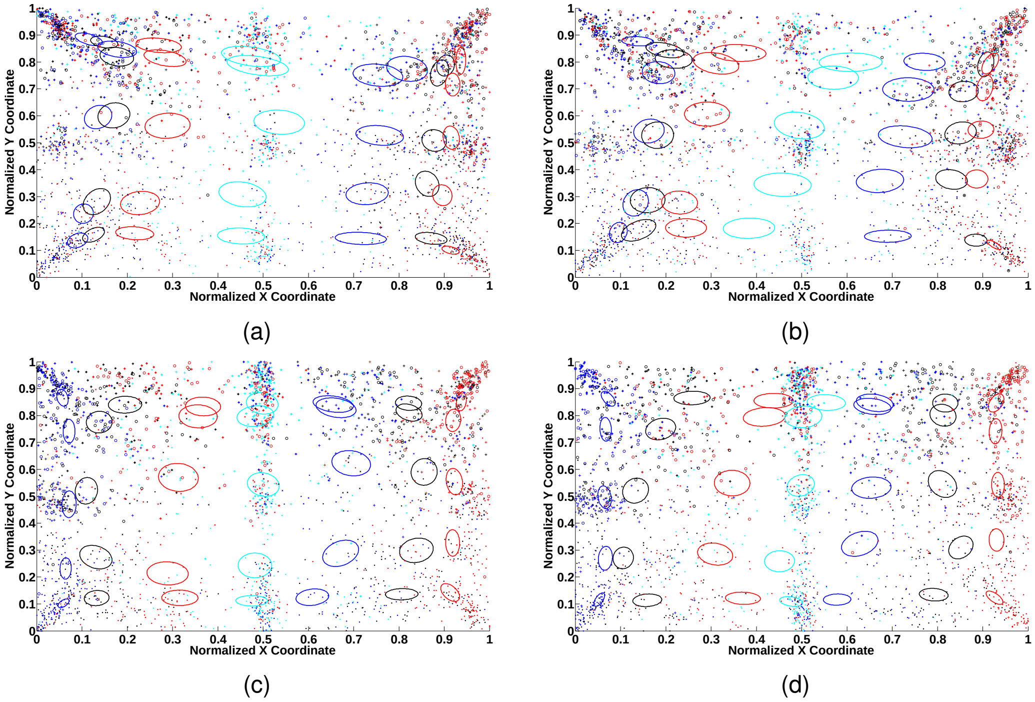
(a) (b)

Fig. 4.5 Screenshots of the application used in User Study 2. (a) training session and (b) main session.

Participants finished one training session and one main session for each experimental condition. In the training session, participants touched a point on a gray square on the tablet and an application rendered corresponding vibrotactile feedback (Figure 4.5(a)). Participants were announced to feel at least 100 vibrations but the number of provided vibrations were not explicitly notified for the sufficient learning. In this training, phantom sensations were rendered regarding to the touched location without the grid limitation. After the training session, participants clicked the ‘Next’ button to proceed to the main session. The main session started after 2 minutes of break. On each trial, participants touched the ‘Start’ button (Figure 4.5(b)) to perceive a phantom sensation in random order and then touched a corresponding location on the gray square. The touched location was presented using a yellow circle, and participants selected whether to press the ‘Next’ button to move to the next trial or to touch another location. The touched locations were estimated by using a resolution of 1250 by 635 pixels.

Participants had 2 minutes of break before starting the session for the next experimental condition. Participants wore headphones that played pink noise to block the faint sound produced by the haptic enchanters.

The data collected in the main sessions and the corresponding location of phantom sensations were screen coordinates in 1230 by 635 pixels, and they were normalized from 0 to 1. By applying an normal distribution model, mean values and their covariance matrices



(a)

(b)

(c)

(d)

Fig. 4.6: All responded locations are plotted for each grid and thirty-five 2D normal distribution plots for each grid are drawn using ellipsoids. Each ellipsoid was plotted using a mean and a covariance matrix of its 2D normal distribution. Each responded location was coded with colors of blue, black, red, cyan, blue, black, and red for its intended target column (1–7) and shapes of dot, x, cross, circle, and asterisk for its intended target row (1–5). (a) PHONE-LINEAR, (b) PHONE-LOGARITHM, (c) RING-LINEAR, and (d) RING-LOGARITHM.

were extracted for each grid point. For an easier representation, a likelihood estimation was processed for each resolution point by using 35 2D normal distribution models.

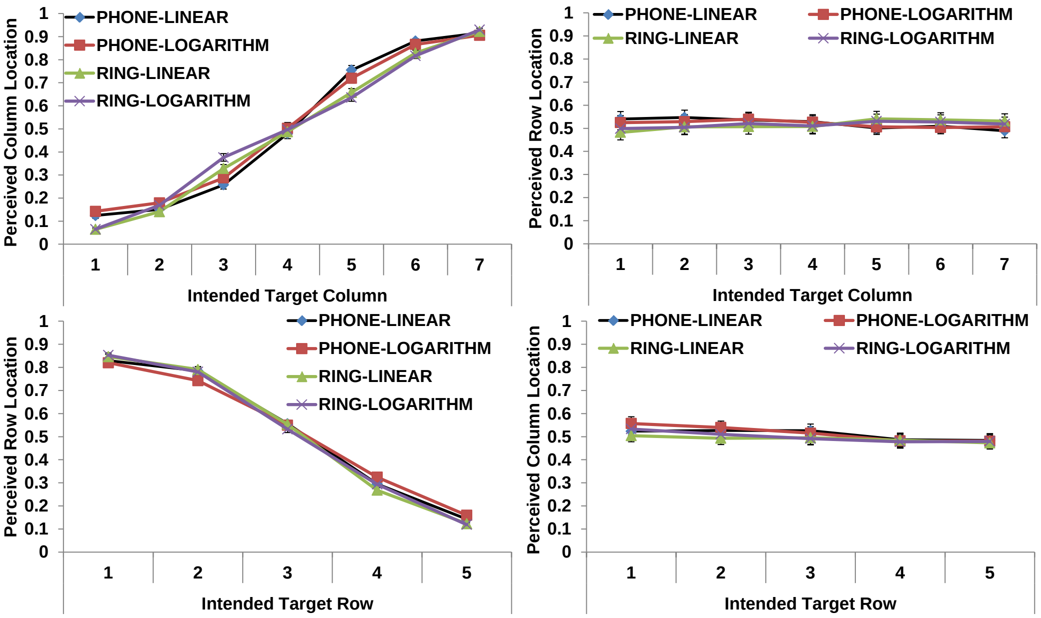
Sixteen participants (3 females and 13 males; 22–31 years old with an average of 26.19) were recruited in this study. No participants reported known sensorimotor disorders, and all participants were right-handed. The experiment took about two hours on average for each participant. Participants were paid about USD 20 for their help after the experiment.

Table 4.1: Analysis Results Using Two-way ANOVA with repeated measures.

PHONE-LINEAR	Factor	F-value	p-value
Perceived row	Row	F(4, 2750)=2310	p < 0.001
	Column	F(6, 2750)=8.73	p < 0.001
	Row:Column	F(24, 2750)=4.31	p < 0.001
Perceived column	Row	F(4, 2750)=11.6	p < 0.001
	Column	F(6, 2750)=1988	p < 0.001
	Row:Column	F(24, 2750)=1.406	p = 0.091
PHONE-LOGARITHM			
Perceived row	Row	F(4, 2750)=1831	p < 0.001
	Column	F(6, 2750)=3.23	p = 0.004
	Row:Column	F(24, 2750)=4.74	p < 0.001
Perceived column	Row	F(4, 2750)=24.4	p < 0.001
	Column	F(6, 2750)=1539	p < 0.001
	Row:Column	F(24, 2750)=3.57	p < 0.001
RING-LINEAR			
Perceived row	Row	F(4, 2750)=2381	p < 0.001
	Column	F(6, 2750)=7.50	p < 0.001
	Row:Column	F(24, 2750)=2.45	p < 0.001
Perceived column	Row	F(4, 2750)=5.39	p < 0.001
	Column	F(6, 2750)=2998	p < 0.001
	Row:Column	F(24, 2750)=2.45	p < 0.001
RING-LOGARITHM			
Perceived row	Row	F(4, 2750)=2684	p < 0.001
	Column	F(6, 2750)=2.77	p = 0.008
	Row:Column	F(24, 2750)=1.83	p < 0.001
Perceived column	Row	F(4, 2750)=21.7	p < 0.001
	Column	F(6, 2750)=2918	p < 0.001
	Row:Column	F(24, 2750)=5.201	p < 0.001

4.2.2 Results

Figure 4.6 shows plots of all responded locations and thirty-five 2D gaussian distribution ellipsoids ($N_i(\overrightarrow{\mu}_i, \mathbf{\Sigma}_i)$, where i is the grid index, $\overrightarrow{\mu}_i$ is a vector of average responded locations of i_{th} grid, and $\mathbf{\Sigma}_i$ is a covariance matrix of i_{th} grid) for each grid point for each experimental condition. Each responded point was coded using color and shape by its



(a)

Fig. 4.7: Normalized average perceived column and row locations versus the intended target column (ITC, bottom) and row (ITR, top) locations. From pairwise t-tests, differences of ITC and ITR on the responded columns and responded rows were statistically significant in all configurations with $p < 0.01$. Statistically significant differences in responded column–ITR and responded row–ITC are not reported here because of small variance values.

column and row indices. In the figure, ellipsoids of row 1–2, column 1–2, and column 6–7 in PHONE and those of row 1–2 in RING overlap. These overlapped ellipsoids imply that the spatial resolution of RING (out-of-body) configuration was finer than that of PHONE (within) configuration. These plots briefly showed the overview of how participants responded to the given 2D phantom sensations (intended target locations), but more information is required to obtain the 2D spatial resolution of phantom sensations.

Average perceived column and row locations were plotted in Figure 4.7. To the intended target column (ITC), users' perceived column (UPC) locations in PHONE shaped like an ogive curve while those in RING were linearly distributed. To the intended target row (ITR), users' perceived row (UPR) locations showed less differences between configurations.

For in-depth analysis, two-way repeated measures of ANOVA was applied to the UPC and UPR and Table 4.1 showed the results. From the analysis, all of UPC and UPR showed statistically significant differences on ITC and ITR in all configurations. Therefore, as a post-hoc test, pairwise t tests with Bonferroni corrections were applied to the all factors. Independent factors of ITC and ITR had statistically significant effects on the UPC and UPR respectively in all configurations ($p < 0.001$). Statistically significant effects were also found in ITC-UPR and ITR-UPC, but the range of perceived locations (0.47–0.52) were far smaller than those in ITC-UPC and ITR-UPR (0.07–0.93) as in Figure 4.7. Therefore, analysis was focused on the ITC-UPC and ITR-UPR.

These statistical analysis can be interpreted that users perceived 7 distinct columns and 5 distinct rows from both of 2D 'out-of-body' phantom sensations (RING) and 2D within phantom sensations (PHONE). However these analysis does not mean that users perceived 35 distinct locations from a 7 by 5 grid yet. To clearly represent the perceptual resolution of 2D phantom sensations, each screen coordinate was color-coded using a grid index with the highest log-likelihood among 35 2D gaussian distributions. First, a log-likelihood of a screen point vector $\overrightarrow{x}$ and i_{th} 2D gaussian distribution $N_i(\overrightarrow{\mu}_i, \mathbf{\Sigma}_i)$ was estimated as follows:

$$ln(L_i(\overrightarrow{\mu}_i, \mathbf{\Sigma}_i|\overrightarrow{x})) = -0.5ln(|\mathbf{\Sigma}_i|) - 0.5(\overrightarrow{x} - \overrightarrow{\mu})^T\mathbf{\Sigma}^{-1}(\overrightarrow{x} - \overrightarrow{\mu}) - ln(2\pi) \quad (4.4)$$

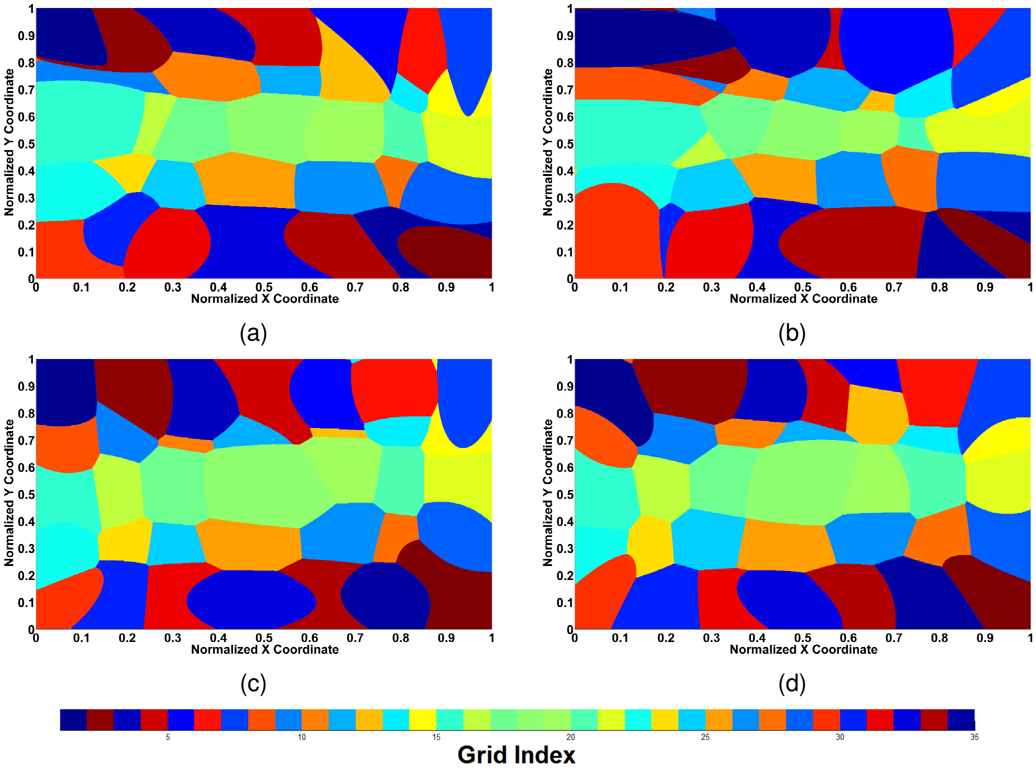
Then the grid index for each screen coordinate was calculated using

$$g(\overrightarrow{x}) = \max_{i \in 1...35} ln(L_i(\overrightarrow{\mu}_i, \mathbf{\Sigma}_i|\overrightarrow{x})) \quad (4.5)$$

Therefore, $g(\overrightarrow{x})$ is the intended target 2D location index of the given screen coordinate $\overrightarrow{x}$.

Figure 4.8 describes color-coded indices of 1235 by 635 screen coordinates calculated by using equation 4.4 and r̃efeq:findingindex. Also, Figure 4.9 shows the percent count of each grid. The standard deviation of them were 1.32, 1.97, 1.33, and 1.11 for PHONE-LINEAR, PHONE-LOGARITHM, RING-LINEAR, and RING-LOGARITHM.

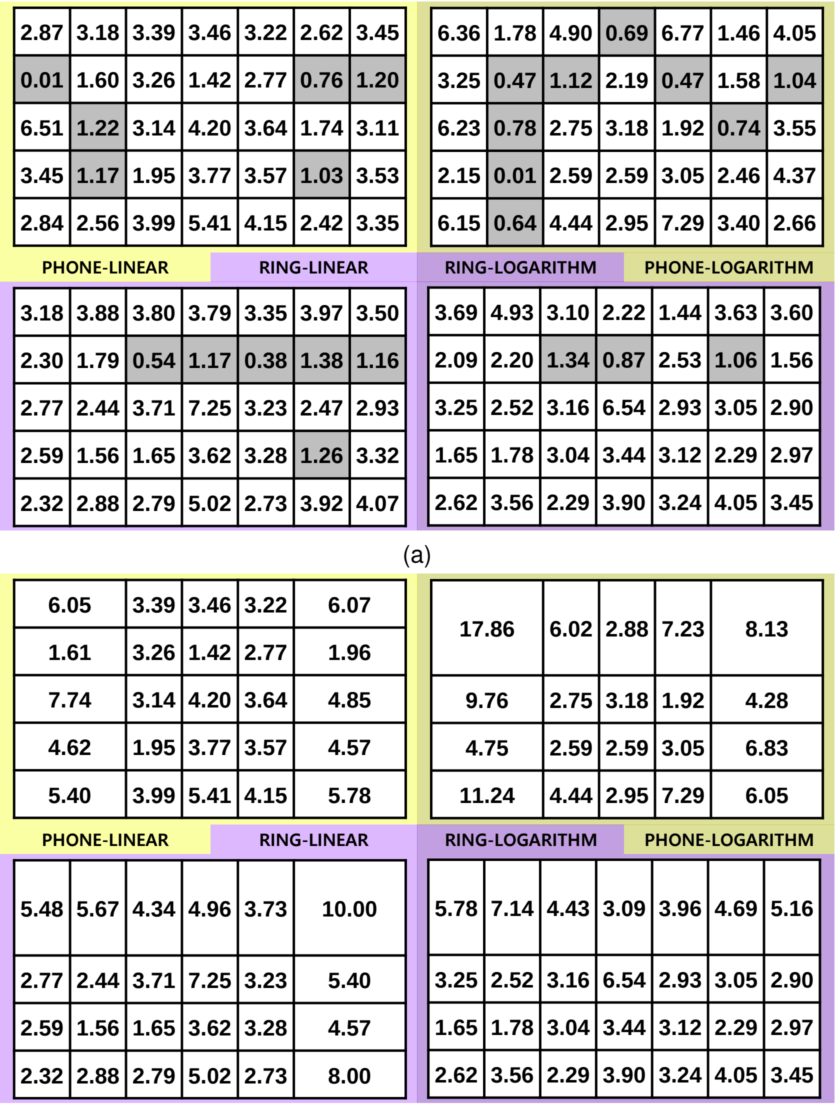
To extract perceptual resolution from results, a guideline of an imperceptible grid location is required. It is clear that RING–LINEAR and RING-LOGARITHM have impercep-



(a) (b) (c) (d) (e)

Fig. 4.8: Color coded responding area to the screen coordinate. Grid index starts from the left-top corner and finishes at the right-bottom corner. An grid index of each screen point was selected with the highest likelihood among 35 2D gaussian distributions. (a) PHONE-LINEAR, (b) PHONE-LOGARITHM, (c) RING-LINEAR, (d) RING-LOGARITHM, and (e) color code of each grid index.

tible grid locations in column 1 – row 2 and column 2 – row 4. In this thesis, half of the mean percent correct value (2.85%; 1 over 35 grid locations) was set as a threshold value and each grid in Figure 4.9 (a) was integrated to make each grid location has percent count values larger than the threshold as Figure 4.9 (b). By using this guideline, perceptual resolutions of 5 by 5, 5 by 4, 6 by 4, and 7 by 4 are recommended when configurations of PHONE-LINEAR, PHONE-LOGARITHM, RING-LINEAR, and RING-LOGARITHM are used.



2.87	3.18	3.39	3.46	3.22	2.62	3.45
0.01	1.60	3.26	1.42	2.77	0.76	1.20
6.51	1.22	3.14	4.20	3.64	1.74	3.11
3.45	1.17	1.95	3.77	3.57	1.03	3.53
2.84	2.56	3.99	5.41	4.15	2.42	3.35

6.36	1.78	4.90	0.69	6.77	1.46	4.05
3.25	0.47	1.12	2.19	0.47	1.58	1.04
6.23	0.78	2.75	3.18	1.92	0.74	3.55
2.15	0.01	2.59	2.59	3.05	2.46	4.37
6.15	0.64	4.44	2.95	7.29	3.40	2.66

PHONE-LINEAR RING-LINEAR RING-LOGARITHM PHONE-LOGARITHM

3.18	3.88	3.80	3.79	3.35	3.97	3.50
2.30	1.79	0.54	1.17	0.38	1.38	1.16
2.77	2.44	3.71	7.25	3.23	2.47	2.93
2.59	1.56	1.65	3.62	3.28	1.26	3.32
2.32	2.88	2.79	5.02	2.73	3.92	4.07

3.69	4.93	3.10	2.22	1.44	3.63	3.60
2.09	2.20	1.34	0.87	2.53	1.06	1.56
3.25	2.52	3.16	6.54	2.93	3.05	2.90
1.65	1.78	3.04	3.44	3.12	2.29	2.97
2.62	3.56	2.29	3.90	3.24	4.05	3.45

(a)

6.05	3.39	3.46	3.22	6.07
1.61	3.26	1.42	2.77	1.96
7.74	3.14	4.20	3.64	4.85
4.62	1.95	3.77	3.57	4.57
5.40	3.99	5.41	4.15	5.78

17.86	6.02	2.88	7.23	8.13
9.76	2.75	3.18	1.92	4.28
4.75	2.59	2.59	3.05	6.83
11.24	4.44	2.95	7.29	6.05

PHONE-LINEAR RING-LINEAR RING-LOGARITHM PHONE-LOGARITHM

5.48	5.67	4.34	4.96	3.73	10.00
2.77	2.44	3.71	7.25	3.23	5.40
2.59	1.56	1.65	3.62	3.28	4.57
2.32	2.88	2.79	5.02	2.73	8.00

5.78	7.14	4.43	3.09	3.96	4.69	5.16
3.25	2.52	3.16	6.54	2.93	3.05	2.90
1.65	1.78	3.04	3.44	3.12	2.29	2.97
2.62	3.56	2.29	3.90	3.24	4.05	3.45

(b)

Fig. 4.9: Percent count of each grid was calculated using maximum-likelihood estimation in Figure 4.8 (a) and the grids were integrated to let each grid location have a percent count value larger than 1.42%. The percent count was estimated by dividing the number of pixels of each grid location by 781,050 pixels (1230 × 635 pixels). A grid location in a uniformly distributed grid has 2.85%. Grid locations below 1.42% were shaded for the convenient recognition.

4.2.3 Discussion

Comparison with the Previous Work

There exist no study that estimated the resolution of stationary 'out-of-body' and within 2D phantom sensations using localized vibrotactile stimulaters. Thus the results of User Study 2 could be a good reference to the literature.

In [53], the authors provided 7 by 5 grid phantom sensations by attaching actuators at the corners of a tablet grabbed by a user with both hands. They estimated the perceptual resolution of the 'out-of-body' stationary 2D phantom sensation using the ANOVA and the percent correct of each grid. Because of non-localized vibrotactile feedback, their perceptual resolution was 5 by 3 where $1_{st} - 2_{nd}$ row, $6_{th} - 7_{th}$ row, $1_{st} - 2_{nd}$ column, and $4_{th} - 5_{th}$ column were not distinct. Compared to their results, the perceptual resolution estimated using ANOVA of intra-hand 2D phantom sensations (PHONE) was 6 by 5 and that of 'out-of-body' sensations (RING) was 7 by 5. This comparison clearly demonstrates that the localized vibrotactile feedback is essential for the clear perception of 2D phantom sensations, however, this approach is not appropriate to estimate the spatial resolution.

Within vs. Out-of-body Phantom Sensations

In User Study 1, the conjecture of the lower estimated IT of the PHONE than that of the RING was the weakened phantom sensations because of the existence of multiple finger joints between the stimulation points on the palm and fingers. The phantom sensation denotes an illusory perception of multiple stimuli as a single stimuli given to a single focal point, therefore if the illusory effect is weakened then a user feels multiple stimuli rather than a single stimulus, or the focus of the single point become vague. Assuming the conjecture of weakened phantom sensation in the PHONE is true, then a hypothesis can be formulated that 'users' response area to phantom sensations in PHONE would be smaller and non-uniform than the area in RING'. This hypothesis seems that it was proven in Figure 4.8 (a) and (b) and Figure 4.9.

Linear vs. Logarithmic Rendering Method

Phantom sensations have been investigated in the literature and most of them reported that the logarithmic rendering method gave more clear perceived locations in 1D [1, 88, 105] and 2D [53]. At the first glance, color coded responding area of RING in Figure 4.8 seemed to support the same conclusion to the literature. However, that of PHONE showed that the linear rendering method was better to render the 2D perceived locations that are uniformly distributed.

The best account of this difference can be formulated as the failure of the perceptual adjustment of distorted amplitudes. First, the logarithmic rendering method of 1D phantom sensations provides higher amplitudes to the locations near the actuator. In the literature, users could discriminate those high amplitudes because they needed to focus only two actuators at a time regardless of the actuator localization. Moreover, 10% of propagated vibrations through a mobile phone also interfered the exact localization. [53] reported that the logarithmic rendering was better in 2D phantom sensations, but their system did not directly attach actuators to users' body and vibration amplitudes were degraded by mass of the tablet. In User Study 2, therefore, the logarithmic rendering in the PHONE overemphasized the four corners and reduced the perceptual accuracy.

Implications to Applications

As the discussion in User Study 1, stationary 2D phantom sensation could be good for secondary or ambient cues guiding the user's attention to the neighborhood of an event of interest in multimodal applications, e.g., for games and entertainment. From the results, the best suggestions of the cues are a 5 by 5 grid for PHONE-LINEAR, a 5 by 4 grid for PHONE-LOGARITHM, a 6 by 4 grid for RING-LINEAR, and a 7 by 4 grid for RING-LOGARITHM.

4.3 Conclusions

This chapter shows a novel contribution of comparison between within and out-of-body phantom sensations as an information transmission channel in User Study 1 and a spatial display system in User Study 2. In User Study 1, estimated IT values without error tolerance were 1.89-2.53 bits for 3 by 3, 4 by 4, and 5 by 5 virtual grids and they were not appropriate to provide a main information. However, estimated IT values with grid-1 error tolerance (2.62 – 3.50) could be an evidence of the potential as a secondary information channel. Thus, User Study 2 investigated near-continuous perceptual resolutions of 2D phantom sensations, and this system could provide at least a 5 by 4 grid spatial cue to a user.

Chapter 5

PhysVib: Physically-Plausible Vibrotactile Feedback Library on Collision Events

Vibrotactile feedback has been used in interactive applications for a variety of purposes [20], for example, to transmit abstract information [66], improve task performance [36], elevate realism [58], and provide enjoyable experience [40]. In such applications, use of the vibrotactile stimuli appropriate to their aims is of paramount importance. Chapter 3 showed the possibility of haptic enchanters as a secondary information channel to the multimodal feedback. Designing vibrotactile stimuli for such feedback, however, is a challenging and time-consuming task and often exacerbated by the lack of suitable authoring software. To alleviate this problem, this chapter presents PhysVib: a software library on the mobile platform extending an open-source physics engine for automatic vibrotactile synthesis upon collision events. PhysVib aims to facilitate mobile application development to the great extent while providing physically plausible vibrotactile feedback. This chapter discusses a library that can automatically generate vibrotactile feedback on collision events and to the collision occurred location, and user studies to prove the performance of PhysVib.

5.1 Structure of PhysVib

Physics engines aim at providing physically accurate simulations while supporting a general class of objects and their interactive behaviors. As a consequence, physics engines have a high computational load, and they are generally designed to satisfy the update rate requirement for graphics, e.g., 30 frame/s. However, haptic feedback requires a greatly higher update rate for the faithful reconstruction of haptic stimuli. Furthermore, physics engines do not provide the simulation results necessary for vibrotactile feedback because they usually treat vibrations as the noise.

PhysVib addresses these challenges by adopting the architecture of multi-rate rendering, where the key complex physical behaviors are simulated at a low update rate and their results are upsampled to obtain a much higher update rate. Multi-rate rendering is an effective platform that allows for an optimal trade-off between simulation accuracy and computational complexity. It has been used in many haptic rendering algorithms involving complex dynamics simulations, e.g., those for deformable bodies [2, 14].

For the implementation, PhysVib adopted AndEngine, an open-source 2D game engine running on Android[1]. AndEngine includes Box2D[2], an open-source 2D physics engine providing essential functions for the simulation of 2D rigid body dynamics. Box2D has been adopted by many more advanced 2D physics engines.

5.1.1 Vibration Generation Model

Consider a 2D virtual environment that consists of many rigid objects. PhysVib assumes that some of the objects are physically connected to the user's hand, working as "haptic cameras" to the virtual environment based on the general virtual window metaphor. Only the collisions associated with haptic cameras trigger vibrotactile feedback. For example, in a driving game, a haptic camera is focused on the car that a user is driving, and collisions that this car makes with any other cars, obstacles, or walls lead to vibrotactile feedback. Another scenario is a user holding a box in which many objects freely move around. If a

[1] http://www.andengine.org

[2] http://box2d.org

haptic camera is set to the box, collisions of the moving objects to the surrounding box are delivered to the user's hand via vibrotactile feedback, but collisions among the moving objects themselves are ignored. Just as out-of-sight visual objects do not need to be rendered, there are no needs to consider the haptic objects that are unlinked to the user's hand. Use of haptic cameras not only preserves the first-person nature of the touch sense, but it also saves the computational load.

If a collision occurs to a camera-focused object, PhysVib uses the following exponentially-decaying sinusoidal model for vibrotactile feedback [72]:

$$x(t) = A \sin(2\pi f t)\, e^{-\tau t}, \tag{5.1}$$

where $x(t)$ is the vibration signal at time t, A is the *normalized* vibration amplitude determined from the impulse magnitudes computed by the physics engine (Section 5.1.3), and f and τ are the natural frequency and vibration decay rate of the camera-focused object, respectively. This is a reality-based model that was empirically confirmed from a variety of real objects by measurement [72]. This model was shown to improve the realism of haptic simulation to the great extent [54], and it has been used for tactile confirmation of touchscreen presses [75]. In two user studies, the perceptual qualities of this model fit to the purpose is estimated (Section 5.2 and 5.3).

If multiple collisions occur between different objects, vibration commands are computed for each collision using (5.1). Then they are added together, and the result is conveyed to the user's hand.

The reality-based vibration model in (5.1) emphasizes the impulse at contact and the material property of the camera-focused object. It takes into account only the first mode of vibration, and this simplicity is beneficial for high-rate synthesis. Moreover, it requires only two additional material parameters (f and τ), minimizing the developer's increased effort in deciding the parameter values of each virtual object (for graphic rendering only, restitution and friction coefficients, density, shape, and damping parameters must be set for each object in Box2D [15]).

More physically accurate synthesis should also consider the shapes of collided objects and the position of collision [24]. This generally results in many higher frequency modes and harmonics in the simulation output. However, a majority of such high-frequency components appear above the upper limit (around 700 Hz–1 kHz) of vibrotactile perceptual bandwidth [20]. Within-band harmonic components often have the amplitude below the corresponding absolute threshold and thus are not perceptible [21]. Even for individually perceptible components, complex vibrations formed by the superposition of multiple frequency components have similar rough sensations regardless of the individual frequencies or amplitudes, not preserving their individual vibrotactile pitches [108]. Therefore, the perceptual merits of more physically accurate synthesis remain dubious at the moment despite its increased computational load. In fact, User Study 1 in this chapter demonstrates that users regard that the reality-based model provides more harmonious vibrations to the visual scene than more physically accurate vibrations synthesized from real contact sounds (Section 5.2).

Although the vibration model is simple, designing and implementing a software library that considers all of the physical plausibility, compelling perceptual quality, computational load, actuator limitation, easiness of use, and general applicability requires great care. How PhysVib deals with these issues are described in the rest of this section.

5.1.2 Extension of Data Structure

Physics engines use different data depending on their target physical system, but they share some common data structures such as geometric and material properties. Extending only the material properties suffices for the purpose so that they include the natural frequencies and the vibration decay rates of objects.

Box2D has two main data structures, called body and fixture, for each object (Fig. 5.1). A body stores essential information for motion such as position, velocity, and mass. Forces, torques, and impulses are applied to bodies. A fixture stores shape (e.g., a polygonal mesh), as well as the coefficients of density, friction, and restitution. One or several fixtures are attached to the body to define the shape and other properties of the object. Collisions between objects are detected by comparing the geometric configurations between the fixtures

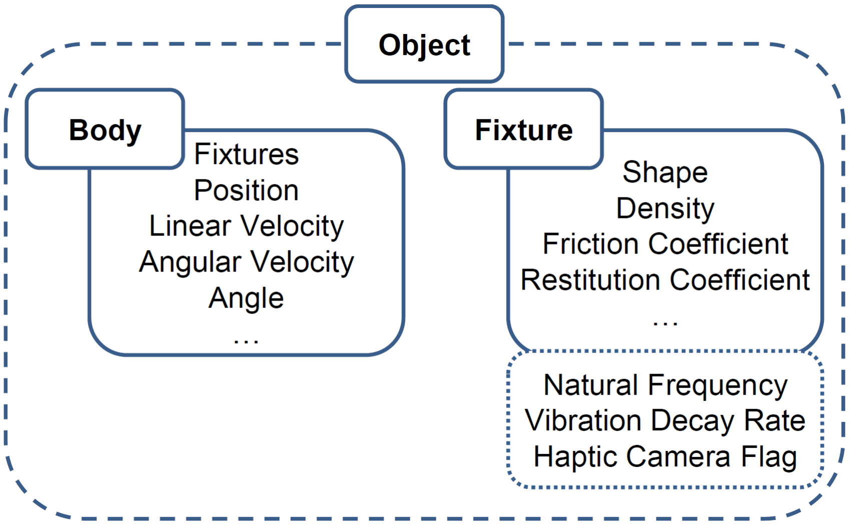


Fig. 5.1 The original data structure of Box2D (solid line) and that extended by PhysVib (dotted line).

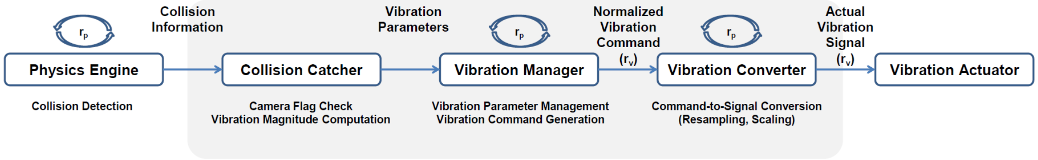

r_p
Physics Engine
Collision Detection
Collision Information
Collision Catcher
Camera Flag Check
Vibration Magnitude Computation
Vibration Parameters
r_p
Vibration Manager
Vibration Parameter Management
Vibration Command Generation
Normalized Vibration Command (r_v)
r_p
Vibration Converter
Command-to-Signal Conversion (Resampling, Scaling)
Actual Vibration Signal (r_v)
Vibration Actuator

Fig. 5.2: Structure and internal processes of PhysVib. r_p: the update rate of Physics engine, r_v: the sampling rate of vibration signal.

of the objects.

A fixture supports a void pointer to add user-defined data. This feature was utilized to support the vibration synthesis model (Fig. 5.1). The extended data structure also includes a flag variable that indicates whether a haptic camera is focused on that fixture. This design enables developers to set the attributes of objects at their creation or to control the haptic camera setting during run time.

To enable automatic vibrotactile rendering, developers set only the three variables of natural frequency, decay rate, and haptic camera flag to the original data of Box2D for each object. The three variables have clear physical definitions and perceptual consequences.

These are an important merit of PhysVib as a software library.

5.1.3 Processes

PhysVib, when executed, creates two processes: a *vibration manager* and a *vibration converter* (Fig. 5.2). The two processes run concurrently with the physics engine at the same update rate r_p. This parallelism guarantees the independence of the two processes from the physics engine used, thereby improving future extendability.

If the physics engine detects collisions, it calls a function named a *collision catcher* and passes collision-related data to the collision catcher. This function selects only haptic camera-focused objects, computes relevant variables for those objects, and sends them to the vibration manager. The role of the vibration manager is to make normalized vibration signals of a high sampling rate $r_v(r_v \gg r_p)$ that are independent of the actuator. At every periodic execution at the update rate r_p, the vibration manager sends segments of the normalized vibration signals to the vibration converter. The vibration converter is responsible for transforming the normalized vibration signals to an actual vibration command and sending it to the actuator. Since this procedure depends on the actuator and communication method used, the vibration converter is designed to be a separate process customizable by the developer. This architecture is designed for efficient multi-rate rendering while maintaining desirable structures for a software library.

Collision Catcher

Box2D calls a callback function using a class ContactListener at the end of each simulation frame if a collision is detected or finished. The collision catcher inherits the ContactListener class. Box2D passes to the collision catcher a fixture pair between which new contact was made or prior contact was broken, i.e., only the changes in the contact state of the virtual objects. The purpose of the collision catcher is to calculate the vibration amplitude A in (5.1) for all the fixtures that are affected by the new collision information.

The collision catcher maintains an undirected graph $G = (V, E)$, which is called a contact state graph, to store and update the contact state between virtual objects. In G, each vertex $v \in V$ represents a fixture of a virtual object. The presence of an edge $e \in E$ between two vertices indicates that the two corresponding fixtures are in contact. At each invocation, the collision catcher inserts new edges to G or delete existing edges from G, reflecting the changes in the contact state. Then, a breadth-first search (BFS) is executed on G from each vertex v that has new contact. This is to build a collision path tree T, which includes all the camera-focused fixtures that are connected to v (Fig. 5.3). The next steps are repeated for each T.

A collision between two fixtures generates an impact force, and it propagates to the collided fixtures if multiple fixtures are in contact. In Fig. 5.3(a), v_1 and v_2 are in direct contact, and the corresponding impact force $\mathbf{F}^r$ (Fig. 5.3(c)) between them can be written as

$$\mathbf{F}^r = \mathbf{F}^r_L + \mathbf{F}^r_A, \tag{5.2}$$

$$\mathbf{F}^r_L = m_1 \Delta\dot{\mathbf{x}}_1 \text{ and } \mathbf{F}^r_A = I_1 \Delta\dot{\theta}_1 \times \frac{\mathbf{l}}{\|\mathbf{l}\|}, \tag{5.3}$$

where m_1, $\Delta\dot{\mathbf{x}}_1$, I_1, $\Delta\dot{\theta}_1$, and $\mathbf{l}$ denote the mass, linear velocity difference, inertia, and angular velocity difference of v_1, and the vector from the contact point to the center of mass of v_1. The impact force to v_2 is $-\mathbf{F}^r$. The impact force is propagated to the contacted fixtures by the principal of action and reaction, and the propagated force moves each fixture. Note that Box2D automatically calculates $\mathbf{x}$ and θ of all fixtures in every time step.

In general, objects connected to each other receive energy from external force (direct or propagated impact force), and this energy is transformed to movement, heat, and vibration. However, the rigid body simulation used in PhysVib considers only movement, assuming zero damping and elasticity of objects, and ignores heat and vibration. This fact necessitates another means for vibration generation. PhysVib selected an approach that embeds an oscillator in each fixture using the reality-based model in (5.1). This method not only provides plausible approximation, but it also allows developers to determine object properties easily—they set only two additional parameters (natural frequency and decay rate) that have clear physical meanings for each fixture.

For a camera-enabled fixture v_i in T, its vibration amplitude A_i is computed by

$$A_i = A^* \frac{\|\mathbf{F}_i^r\|}{m_i + m_h}, \tag{5.4}$$

where A^* is a normalization constant, m_i is the mass of v_i, and m_h is the mass that represents the effective mass of the user's hand touching the object (Section 6.2.3 describes how to determine A^* and m_h). This oscillator model is consistent with [72], which experimentally demonstrated that the vibration amplitude is approximately linear to the impact velocity; compare (5.2), (5.3), and (5.4). The model also reflects the loading effect of the user's hand appropriately.

An important issue here is the computation of $\Delta\dot{\mathbf{x}}_i$ and $\Delta\dot{\theta}_i$ for (5.2) and (5.3). Two general approaches can be used for that purpose. One is to predict the velocities at the initiation of a collision and start vibrotactile feedback immediately. The vibrotactile feedback command is updated during the collision whenever new information about $\Delta\dot{\mathbf{x}}_i$ and $\Delta\dot{\theta}_i$ is available. However, this approach suffers from prediction errors, and this problem can be exacerbated when multiple collisions occur. The other approach, which PhysVib follows, is to await the release of the contact, and then compute $\Delta\dot{\mathbf{x}}_i$ and $\Delta\dot{\theta}_i$ and generate a vibration. This approach allows for the best accuracy, but it may incur a *crossmodal* delay from the time at which the user visually detects a collision to the time at which the user begins to feel a vibrotactile response. If the crossmodal delay is imperceptible, this approach is inarguably superior in terms of simulation accuracy and algorithmic simplicity.

To determine which approach to use, the crossmodal delay of PhysVib was investigated. Physics engines for rigid body dynamics should break contact ideally in one time step of simulation, but Box2D adopts an impulse-based approach that takes several steps from the start of a collision to its end. The contact duration was measured for the bouncing objects used in User Study 1 (Section 5.2), and PhysVib took from 10 to 50 ms to break contact regardless of the material properties of the objects. In general, this small crossmodal delay is imperceptible to users. For example, a crossmodal delay from 18 to 72 ms imposed on a simple typing task on a touchscreen showed similar perceptual effects and task performance to those of 18 ms delay [48]. Further, crossmodal delay is more difficult to notice when the

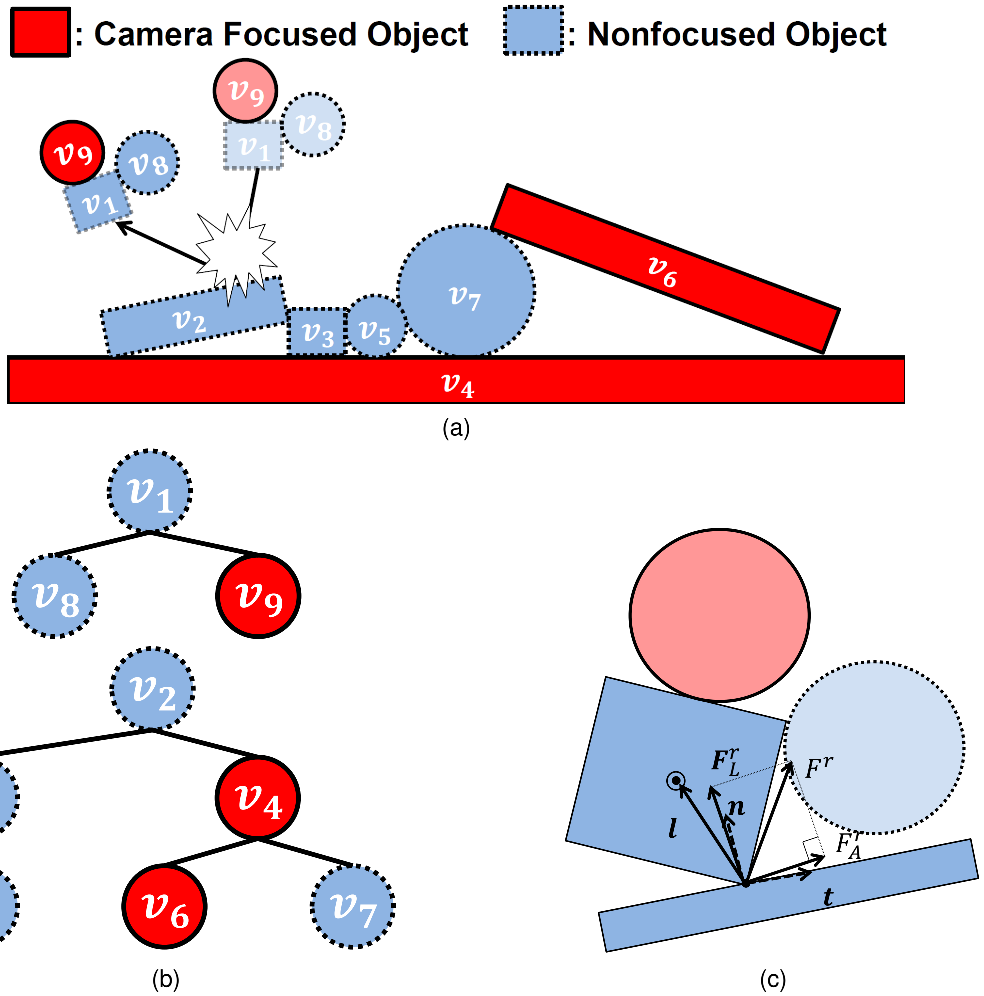


Fig. 5.3 Building collision path trees using breadth-first search. (a) A new collision is made between v_1 and v_2 among multiple fixtures. (b) Two collision path trees are built with v_1 and v_2 as the root, respectively. (c) Impact force $\mathbf{F}^r$ is generated from the collision between v_1 and v_2. $\mathbf{F}_L^r$: linear impact force, $\mathbf{F}_A^r$: angular impact force, $\mathbf{n}$: surface normal vector, $\mathbf{t}$: surface tangent vector, and $\mathbf{l}$: the vector from the contact point to the center of mass of v_1.

virtual environment becomes complex and immersive alongside the user's frequent motor activity (expected in the use of PhysVib; see Fig. 5.5 for examples). Therefore, PhysVib computes $\Delta\dot{\mathbf{x}}_i$ and $\Delta\dot{\theta}_i$ after the release of collision. This method well preserves the perceptual synchrony between the visual and tactile detection of contact.

The collision catcher repeats the above procedure for all haptic camera-focused fixture pairs in contact. As a result, it makes a linked list L_{col} in which each node contains f, τ, and A for the corresponding fixture pair. L_{col} is then passed to the vibration manager for vibration command generation.

Vibration Manager

Unlike the event-driven collision catcher, the vibration manager runs in parallel with the physics engine at the same update rate r_p. The vibration manager maintains a linked list L_{vib} to store parameters for vibration generation. Whenever L_{col} is passed from the collision catcher, a node is created in L_{vib} for each new pair of fixtures in L_{col}, and f, τ, and A are copied into it from the corresponding node of L_{col}.

Each node in L_{vib} contains two more variables of vibration initiation time t_0 and duration t_d. t_0 is the system time at which L_{col} is passed to the vibration manager. t_d is determined by estimating the time interval after which the exponentially-decaying vibration command $x(t)$ in (5.1) becomes perceptually negligible for this collision. t_d is chosen as the minimum t such that

$$e^{-\tau t} < \gamma, \tag{5.5}$$

where $\gamma < 1$ is a user-defined parameter.

Using L_{vib}, the following procedure is executed at every periodic update of the vibration manager. First, L_{vib} is scanned to find the nodes that satisfy $t > t_0 + t_d$. Such nodes no longer produce significantly large vibrations, so they are deleted from L_{vib}. Second, for each node that remains in L_{vib}, the vibration manager uses (5.1) with the parameters of the node to compute

$$X = \left\{ x\left(t + \frac{1}{r_v} i\right) \middle| 0 \le i < \left\lceil \frac{r_v}{r_p} \right\rceil \text{ for integer } i \right\}, \tag{5.6}$$

where r_v is the sampling rate of the normalized vibration command (5 kHz by default). X represents a vibration waveform for $(1/r_p)$ s (the update period of the vibration manager) sampled at r_v Hz for the collision represented by the node. Third, all Xs are added into a superimposed signal and if it exceeds the amplitude bound of $[-1, 1]$ then it is normalized.

This is to preserve all vibration components from the multiple collisions while abiding by the input range of the actuator. Last, the superimposed vibration signal is sent to the vibration converter.

The vibration manager corresponds to a signal interpolator that upsamples the slow simulation results of a physics engine in a multi-rate rendering architecture. The interpolation makes use of the reality-based vibration model for plausible contact simulation. In this regard, PhysVib is a hybrid of physically-based and data-driven rendering.

Vibration Converter

The role of vibration converter is to transform the normalized vibration command to the actual vibration signal that is appropriate for the vibrotactile actuator used and then send it to the actuator. To this end, the normalized command with the sampling rate r_v needs to be resampled considering the actuator dynamics and the communication bandwidth. The normalized command should also be re-scaled to fit the input range of the actuator. Hence, the vibration converter should be overridden by application developers.

For example, to use a default actuator in a smart phone through a built-in Android function, a normalized vibration command needs to be quantized to a binary signal indicating only on and off of the actuator. Alternatively, a vibration command can be sent to an external wideband actuator through the audio channel of Android after upsampling to the audio rate (44.1 kHz) and suitable amplitude scaling. This method is simple and allows exact synchronization between audio and vibrotactile signals when audio feedback is also used. This method was used in User Study 1 and 2. Examples of the two methods are presented in Fig. 5.4.

5.1.4 Selection of Parameters

To obtain high-quality simulation results, a few parameters need to be chosen with care. The first parameter is the update rate r_p of physics engine. Since most physics engines for mobile devices use discrete collision detection algorithms, using low r_p increases the probability to miss collisions, letting fast objects in colliding paths pass through. This is

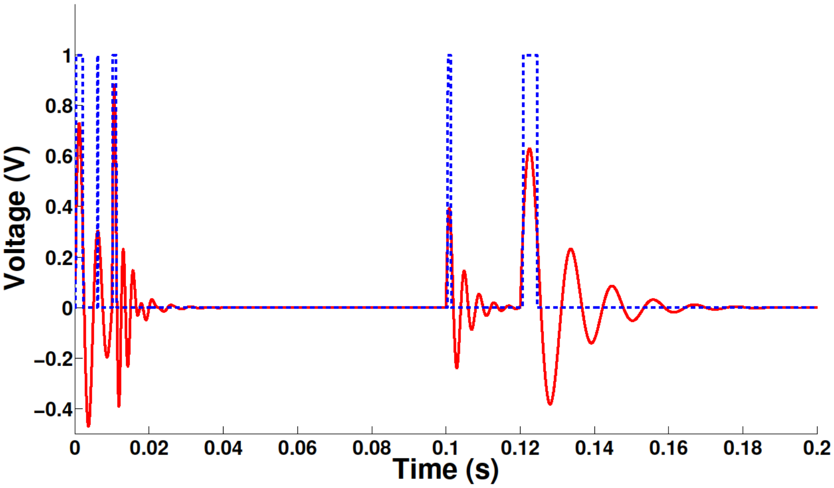
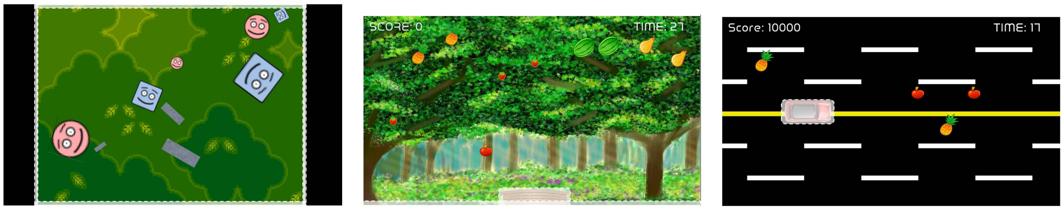


Fig. 5.4 Vibrotactile command for a wideband actuator (red solid line) and for Android default function (blue dotted line).



(a) (b) (c)

Fig. 5.5: Three exemplar applications developed using PhysVib for demonstration. Objects highlighted with white boxes are enabled for vibrotactile feedback. (a) Shake the box. (b) Fruit basket. (c) Driving.

inevitable unless more sophisticated but much more complex and slow continuous collision detection algorithms (e.g., [78]) are used. Increasing r_p improves this problem but intensifies the computational load. Developers must choose an optimal value of r_p depending on the physics engine, the application (object size, maximum velocity, and so on), and the computing power being used. PhysVib uses $r_p = 100\,\text{Hz}$ as the default value.

Second, the default sampling rate r_v is set to 5 kHz to handle normalized vibration commands. This value is approximately ten times of the upper bound of vibration frequencies that can be mediated by the PC (Pacinian) channel [9]. Using a sampling rate that is 10–20 times higher than the signal bandwidth is generally sufficient for faithful signal reconstruction [28].

Third, the normalization constant A^* of vibration amplitude and the observer's hand mass m_h in (5.4) need to be determined for each application. Following models are used to make the amplitude of the strongest collision less than 1 in most applications:

$$A^* = \frac{1}{2v_{max}} \quad \text{and} \quad m_h = m_{max}. \tag{5.7}$$

v_{max} is the maximum velocity of objects in the environment and largely depends on the application and user interaction methods. PhysVib estimates $v_{max} = WD$ as a default value, where W is the screen width in pixels and D is the distance in the environment per screen pixel, both in the logical coordinates. This corresponds to the velocity change when an object moves from one end of the device screen to the other end in 1 s. m_{max} is the greatest mass of all bodies in the environment. It is automatically computed by PhysVib at start or when objects are added or deleted. Users have a full control of A^* and m_h any time.

Last, γ in (5.5) has the role of pruning out perceptually insignificant residual vibrations to lessen the computational load. Using a large value for γ is advantageous for computation, but it may invoke a discontinuous sensation when the vibration is terminated. Thus, an optimal value for γ should be chosen considering the computing power and the actuator performance. The default value is $\gamma = 0.1$.

In practice, once good r_p, r_v, and γ are found for the computing platform and the actuator used, there exists little need for tuning them further for each application. A^* and m_h can be either automatically decided using (5.7) or manually set depending on the application.

5.1.5 Examples

To demonstrate the versatility of PhysVib as a general software library, three example applications were made using PhysVib. Example 1 shown in Fig. 5.5(a) is built upon the "shake the box" metaphor, where a number of objects with different shapes, sizes, masses, and material properties move around while making collisions among themselves and to the walls.

A haptic camera is focused on the walls, resulting in a similar environment to those presented in [85, 104]. In Example 2 (Fig. 5.5(b)), fruits fall from trees, and the user controls the position of a basket to catch the fruits into the basket. When a fruit collides with the basket, it triggers vibrotactile feedback with its property determined based on the material, mass, and velocity of the fruit. Example 3 in Fig. 5.5(c) is a classical driving game where a haptic camera is focused on the car. When the driver's car hits obstacles on the road, various vibrotactile effects are provided depending on the obstacles and the car velocity. The above examples were implemented using the default values of r_p, r_v, and γ and the default rules of deciding A^* and m_h in (5.7).

5.1.6 Advantages

PhysVib inherits the general merits of physics engines: saving application development time to the great extent by providing the functionality of automatic vibration synthesis to collision events. This is especially beneficial to developers who are unfamiliar with the technical and perceptual principles of vibrotactile feedback. In particular, the design of PhysVib using the reality-based vibrotactile feedback model requires to define only the two additional parameters, natural frequency and vibration decay rate, for each graphical object. Additionally, since PhysVib uses common features of physics engines such as callback functions and user-customizable data, it can be easily integrated with other popular physics engines such as Bullet, Farseer, and Chipmunk. PhysVib is open to the public and available for download at [*https://github.com/maharaga/PhysVib*].

The next step is to verify the perceptual quality of vibrotactile feedback enabled by PhysVib. The following two sections report two user studies.

5.2 User Study 1

A collision between objects produces both a sound and a mechanical vibration. In PhysVib, vibrations are modeled using the reality-based model that considers only the first mode of vibration. Although this model well accounts for contact vibrations generated from real objects [72], the sound produced by the same event exhibits a much more complex waveform. Thus, one might argue that converting contact sound to vibration is more physically accurate and leads to better perceptual quality. However, there also exist the technical and perceptual grounds that limit the effectiveness of the sound-to-vibration approach, as discussed earlier in Section 5.1.1. The aim of User Study 1 was to compare the user-perceived value of vibrotactile feedback between the reality-based model and the sound-based synthesis.

5.2.1 Methods

Participants

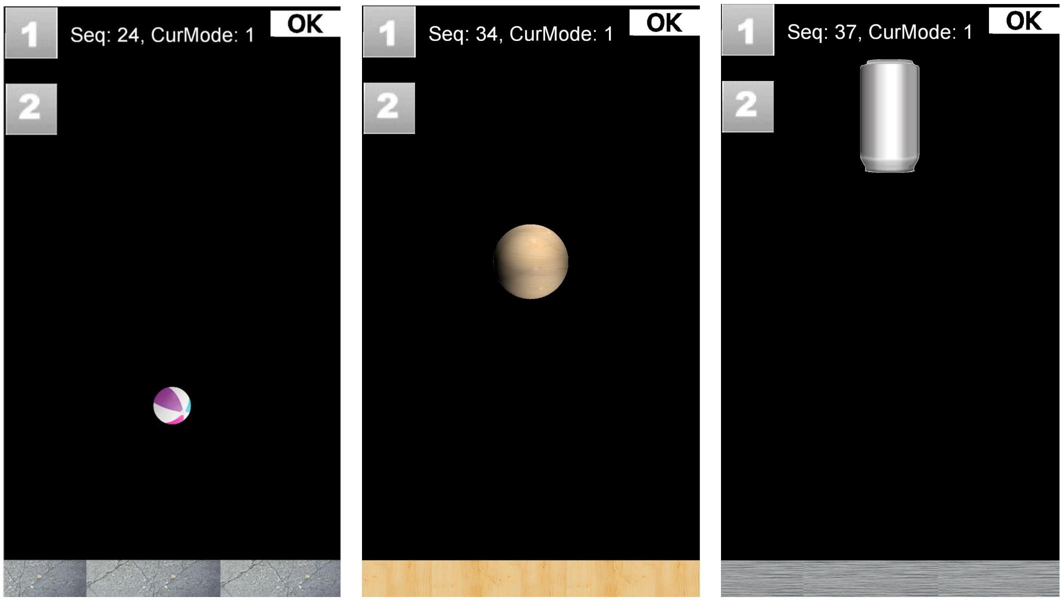
Twenty participants (15 male and 5 female; 2 left-handed; mean age 23.1 years) participated in this user study. They had no significant previous exposure to diverse vibrotactile effects except for their regular use of mobile phones or game pads. They were informed of experimental procedure and conditions using a written document prior to the study and then signed on a consent form. Participants were compensated for their help with 10,000 KRW ($\simeq$ 10 USD).

Apparatus

A commercial smart phone (Samsung SHW-M250S) was used in this study. To provide wideband vibrations, an external actuator (TactileLabs Haptuator 2; input range ± 3 V) was attached to the back panel of the phone. This actuator has resonance at around 110 Hz and a flat magnitude response between 300–800 Hz. It was the most appropriate one preserving the mobile context of this research when this work was done. The actuator was powered by a custom amplifier through the sound output of the phone.

Experimental Conditions

Three pairs of virtual objects were used in this study. Each pair consisted of one dynamic object and one stationary large object. The dynamic object made a free fall to the stationary object. When a collision was made, vibrotactile feedback was provided using either the reality-based model (RM) or the sound-to-vibration approach (SV). The three object pairs



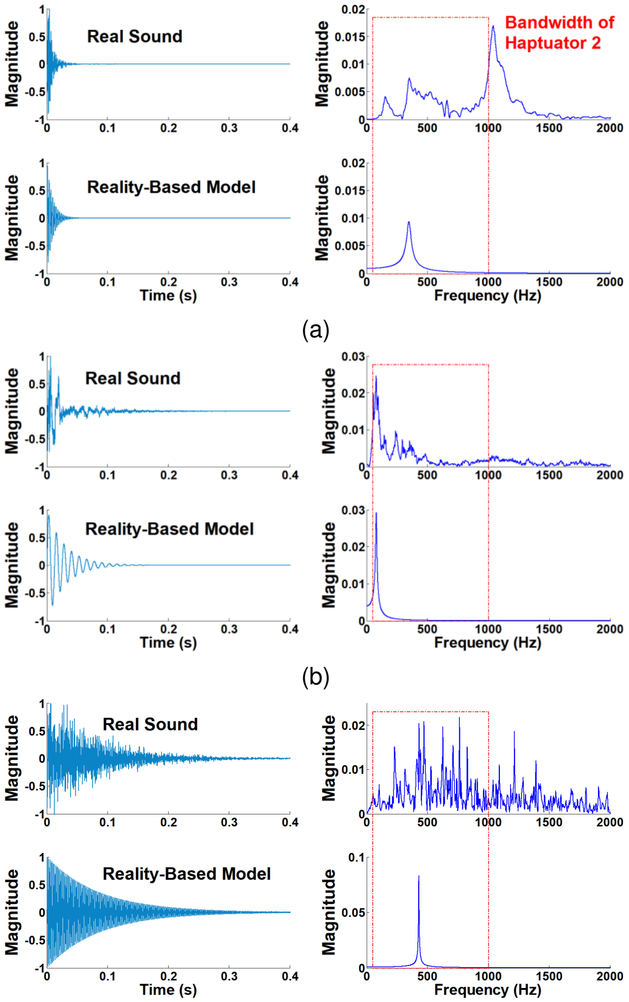
(a) (b) (c)

Fig. 5.6 Screen shots from User Study 1. (a) Rubber ball–concrete block, small size, and 1 m falling distance. (b) Wooden sphere–wooden board, medium size, and 2 m falling distance. (c) Steel can–steel plate, large size, and 3 m falling distance.

were: a hollow rubber ball and a concrete block, a wooden sphere and a wooden board, and a hollow steel can and a steel plate (Fig. 5.6). The purpose of using free-falls was to allow participants to concentrate on perception only, not interfered with voluntary movement for interaction.

Contact sounds for the three object pairs were downloaded from a commercial database (`www.soundsnap.com`). Their normalized waveforms and magnitude spectrums are shown in Fig. 5.7. The natural frequency f of the reality-based model was taken as the frequency of the highest peak within the actuator bandwidth in the spectrum of the corresponding contact sound. Its exponential decay rate τ was estimated by fitting the first and second highest peaks in the time domain to the envelope function ($e^{-\tau t}$ in (5.1)).

Since the momentum depends on the mass and the velocity difference of collided objects, the size and falling distance of a free-fall object were also varied as independent variables. For each object pair, three sizes (small, medium, and large) of dynamic objects were tested.



(a)

(b)

(c)

Fig. 5.7 Sound and vibration stimuli used in User Study 1. Left plots show time-domain waveforms, and right plots show their magnitude responses. The bandwidth of the actuator used is represented by a red dotted line in the right plots. (a) Rubber ball–concrete block. (b) Wood sphere–wood board. (c) Steel can–steel plate.

Their dimensions are specified in Table 5.1, along with other material parameters. The falling distance was the length from the lower end of a falling object to the upper end of a stationary object. It was one of 1, 2, and 3 m for each object pair. The gravity constant used for physics simulation was 9.8 m/s^2.

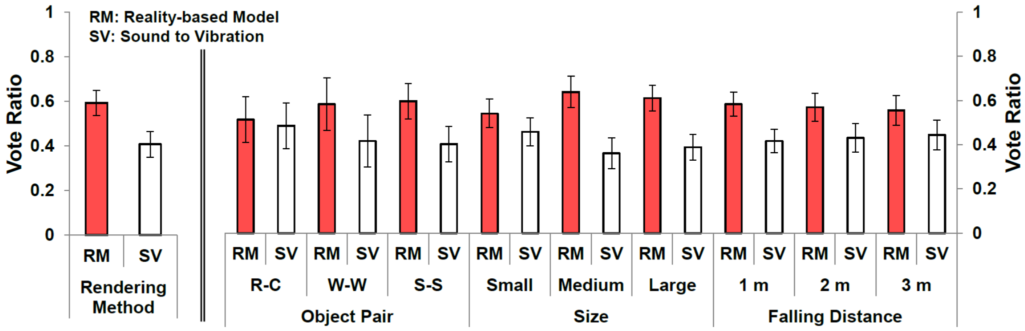


Fig. 5.8: Ratios of the medians of votes made to the two vibration rendering methods. In the left plot, all the data are pooled. In the right three plots, the data are broken down to see the effect of object pair, dynamic object size, and falling distance. R-C: Rubber ball–Concrete block. W-W: Wooden sphere–Wooden board. S-S: Steel can–Steel plate. Error bars represent 95% confidence intervals.

Vibration feedback for RM was provided using PhysVib. The default values were used for r_p, r_v, and γ regardless of the object pair. A^* and m_h computed using the default rules in (5.7) are listed in Table 5.1 for each object pair. The scaling constant used in the vibration convertor was 3 V for the actuator used. To render vibrations using SV, its amplitude was computed using PhysVib, i.e., A_i in (5.4), and then multiplied to the corresponding sound waveform shown in Fig. 5.7. Then the resulting waveform was played back in the same way of that of RM.

One problem was that a tactile vibration produced by SV is clearly perceived weaker than the corresponding vibration generated by RM. It is because when two vibrations with different frequencies are superimposed, the resultant vibration feels weaker than the individual vibrations [58]. To prevent this perceptual intensity difference from affecting experimental results, the perceptual intensities between RM and SV were equally adjusted. For each of the 27 experimental conditions (3 object pairs × 3 sizes × 3 falling distances), the scaling constant was estimated for RM that resulted in the perceptually equally strong vibration to that produced by SV. This process was done using the method of adjustment with three expert participants. These scaling constants were multiplied to the RM vibrations.

Table 5.1: Parameters used for simulation in User Study 1.

Dynamic Object	Rubber ball	Wood sphere	Steel can
Size on screen (mm)	3, 6, 9 (radius)	3, 6, 9 (radius)	3×6, 6×9, 9×12
Size in simulation (m)*	0.15, 0.30, 0.45 (radius)	0.15, 0.30, 0.45 (radius)	0.157×0.30, 0.314×0.60, 0.471×0.90
Density (kg/m^3)**	120	850	1193
Mass (kg)	8.48, 33.92, 76.32	59.08, 238.73, 537.15	56.71, 226.85, 510.41
Restitution coefficient	0.688	0.603	0.597
Stationary Object* **	**Concrete block**	**Wooden board**	**Steel plate**
Size on screen (mm)	58 × 6	58 × 6	58 × 6
Size in simulation (m)*	2.9 × 0.3	2.9 × 0.3	2.9 × 0.3
Density (kg/m^3)	1000	850	7820
Mass (kg)	870	739.5	6803.4
Restitution coefficient	0.500	0.603	0.597
Natural frequency (Hz)	80	349	430
Vibration decay rate	34.13	106.99	11.97

* Object sizes in Box2D must be between 0.1 and 10 m. Otherwise, simulation results tend to become numerically unstable.

** The rubber ball and the steel can are hollow and have thicknesses of 10% of their size. Their densities were calculated using the material densities of 1200 kg/m^3 (rubber) and 7820 kg/m^3 (steel).

*** Haptic focus was on the stationary objects. Their masses were used as m_h in (5.7).

Procedure

On each trial, participants chose which vibration to perceive by touching one of the two top-left buttons that were labeled 1 and 2 and displayed on the touchscreen of the smart phone; see Fig. 5.6 for the user interface (UI). Then the dynamic object fell, making a vibration at contact with the stationary object using the designated rendering method. Participants were asked to repeat this procedure by selecting the two buttons alternately and judge which vibration was perceived *more harmonious to the visual collision event*. It was expected that participants would understand the physical properties of collisions by visually perceiving the size, material, and velocity of objects and rely on this information to decide whether vibrotactile feedback is physically plausible. After making a decision, participants pressed the corresponding number button and then the OK button on the top-right corner. This finished the trial, and the next trial started immediately.

Participants grasped the smart phone in their non-dominant hand and controlled the touchscreen interface using their dominant hand. They wore headphones that played pink noise to block the effect of sound produced by the vibration actuator.

The experiment consisted of 108 trials (3 object pairs × 3 dynamic object sizes × 3 falling distances × 2 button configurations × 2 blocks). In one trial, the SV stimulus was assigned to Button 1 on the UI, and the RM stimulus was to Button 2. In the other trial, the button configuration was switched. The order of 54 trials in each block was randomized for each participant. Participants took a 3-min rest between the two blocks.

After completing the main experiment, participants had an interview and wrote down on a document how they distinguished different vibrations and evaluated the harmony of a vibration to the visual event. The entire experiment took 45 min on average.

5.2.2 Results and Discussion

Fig. 5.8 shows the ratios of participants' votes made to the two vibration rendering methods for the three independent factors of object pair, dynamic object size, and falling distance. In all the cases, RM was selected more frequently than SV for conveying a more harmonious sensation to the visual collision event. When all the data were pooled, RM garnered 59% of votes while SV received 41%. This difference was statistically significant (Wilcoxon signed-rank test; $W = 412.5$, $p = 0.017$). Among the three independent factors, only dynamic object size had a statistically significant effect on the number of votes (Friedman test; $\chi^2(2) = 6.68$, $p = 0.036$). Object pair and falling distance did not ($\chi^2(2) = 0.078$, $p = 0.96$; $\chi^2(2) = 0.10$, $p = 0.95$). A post-hoc test (Mann-whitney test with Bonferroni corrections) was applied on dynamic object size, but it did not find any pair with a significant difference.

Although not statistically significant, two object pairs, a wooden-sphere and a wooden board (W-W) and a hollow steel can and a steel plate (S-S), showed a larger vote difference than a hollow rubber ball and a concrete block (R-C). This is likely due to the more dissimilar waveforms of W-W and S-S between RM and SV compared to those of R-C (Fig. 5.7). This means that when two waveforms were easier to distinguish, participants tended to choose RM more often. A vibration that includes multiple spectral components conveys a sensation of high dissonance and roughness [108]. No other clear differences between SV and RM could be found in their time-domain waveforms (Fig. 5.7). In fact, many participants commented that they preferred clear and pleasant vibrations than rough and dissonant vibrations in the post-experimental survey.

The vote ratios shown in Fig. 5.8 do not exhibit clear patterns for the effects of dynamic object size and falling distance. Hence, the vote ratios were examined for the nine combinations of the two independent variables. Fig. 5.9 shows that the median vote ratios for RM were close to 50% (the chance probability) when the dynamic object size was small for all the falling distances. For the medium- and large-size dynamic objects, the median vote ratios for RM were more similar to the grand mean (59%). This is consistent with the participants' unanimous report that the two vibrations produced by RM and SV were difficult to discriminate when a small dynamic object was included. It appears that participants could distinguish the two kinds of vibrations when the vibration amplitude was sufficiently large for the medium and large dynamic objects and also preferred the vibrations generated by RM. It could be because RM's vibrations invoke clearer sensations.

In conclusion, the reality-based model provided more congruent vibrotactile feedback to visual collision events than the sound-based synthesis with mild yet statistically significant differences. Therefore, it is safe to assert that the user-perceived value of the reality-based model is not inferior to that of sound-based vibration feedback for vibrotactile rendering of contact. This is in addition to the greatly lower computational complexity of the reality-based model than that of sound synthesis algorithms (see Section 2.3).

5.3 User Study 2

PhysVib was designed to provide vibrotactile feedback that feels plausible to the collision events visually displayed on the screen. User Study 1 demonstrated that the reality-based

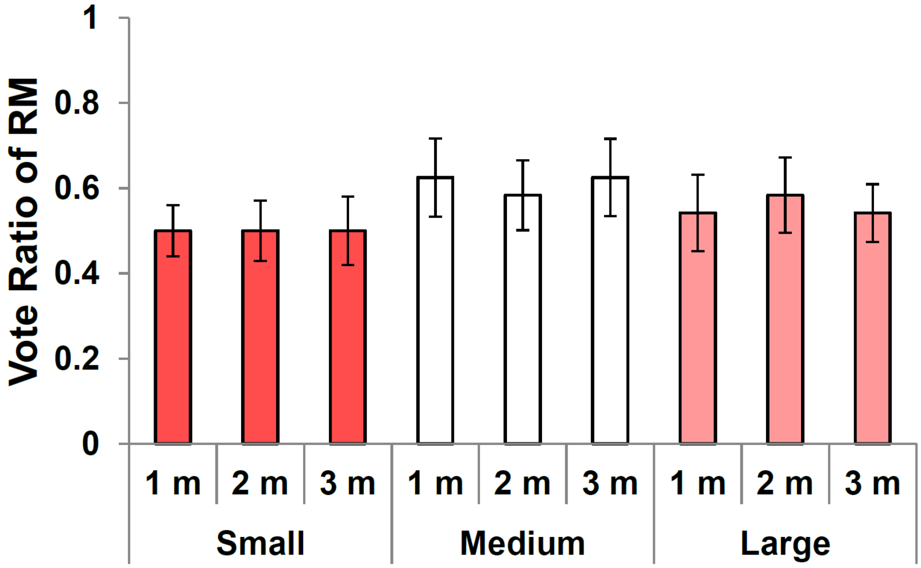


Fig. 5.9 Median vote ratios for the reality-based model for size and falling distance. Error bars indicate 95% confidence intervals.

model used in PhysVib is viable for the purpose. In User Study 2, the extent to which PhysVib improves user experience was estimated compared to other simpler forms of vibrotactile feedback including two methods used in prior work [85, 104].

5.3.1 Methods

Participants

Twenty participants (15 male and 5 female; all right-handed; average age 22.6 years) took part in this user study. All participants reported no known sensorimotor impairments. Participants read written instructions and signed on a standard consent form before the experiment. Each participant was paid 20,000 KRW ($\simeq$ 20 USD).

Apparatus

The mobile phone and vibration actuator used in User Study 1 was also used in this experiment to provide vibrotactile feedback-enabled dynamic simulations. Participants used another phone (LG Optimus G Pro) to enter responses.

Experimental Conditions

This user study employed the “shake the box” application (Fig. 5.10). Participants interacted with a bounding box representing the mobile phone, while experiencing vibrotactile feedback when dynamic objects collided to the box. The application displayed nine dynamic objects consisting of three small, three medium, and three large objects. The material parameters of the dynamic objects and the bounding boxes were the same as in Table 5.1. The sizes on screen of the dynamic objects were reduced to 66% so that they could move around easily within the bounding box. The size of the bounding box used for simulation was 6.4×3.6 m with thickness 0.02 m for all the three object pairs. The mass of the bounding box was 400 kg for a concrete block, 340 kg for a wooden box, and 3128 kg for a steel box. A haptic camera was set to the bounding box.

The application supported two general interaction methods: touch-and-drag and tilting. A touch-and-drag gave an impulse vector to the touched dynamic object. The direction of the impulse was the same as the dragging direction, and its magnitude was proportional to the dragging distance with a coefficient of 32 kN·s/m. All objects were exposed to a gravity field of 9.8 N·m/s^2, and tilting changed the direction of gravity accordingly. The application used the default parameters of r_p, r_v, and γ and the default rules for A^* and m_h.

Eight different forms of vibrotactile stimuli were tested. At one end of the stimulus continuum, the stimulus used the full functionality of PhysVib; the amplitude, frequency, and decay rate were varied depending on the parameters of collided objects and the simulation results of the physics engine. At the other end, the stimulus was an impulse-like, clearly perceptible rectangular pulse, similar to those used in [85], without reference to object properties or the physics involved. Its parameters were 1.5 g for amplitude, 250 Hz for frequency, and 20 ms for duration. This pulse was found to elicit the best subjective responses in pilot experiments (also see [75]). The other six vibrotactile stimuli were between the two extremal stimuli in a factorial design of amplitude, frequency, and decay rate—see Fig. 5.12 for their waveforms and notations. Note that the AF-FF-DRV method is similar

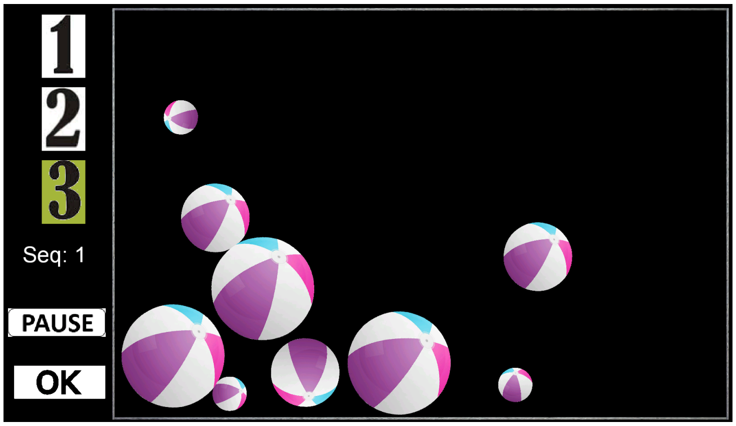
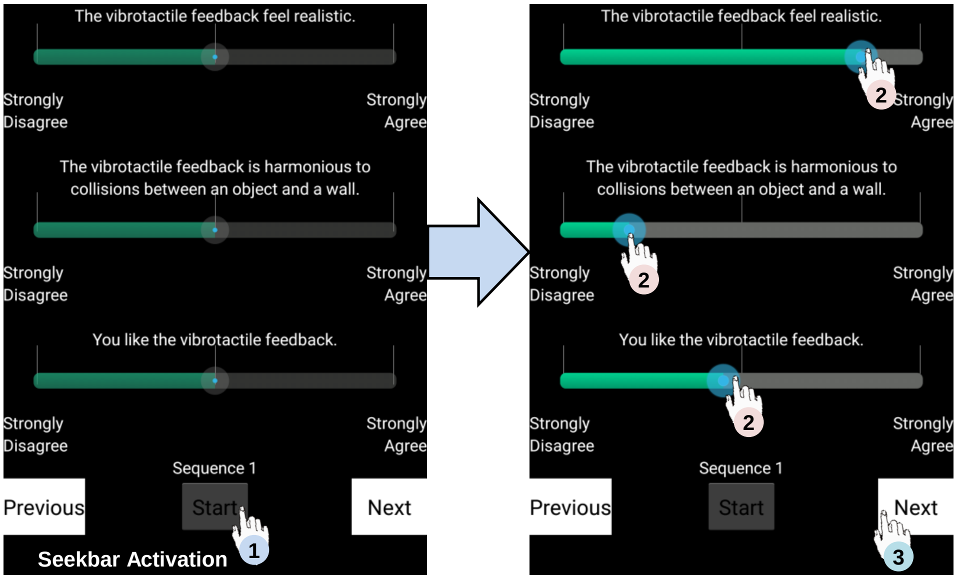


Fig. 5.10 Screenshot of a physics simulation program used in User Study 2, showing a rubber ball–concrete wall pair.


The vibrotactile feedback feel realistic.

The vibrotactile feedback is harmonious to collisions between an object and a wall.

You like the vibrotactile feedback.

Fig. 5.11 Screenshots of a UI that participants used to enter responses. (Left) The initial screen. Participants pressed the start button to activate seekbars. (Right) After moving the seekbar cursors, participants touched the next button to save their responses.

to the stimuli used in [104].

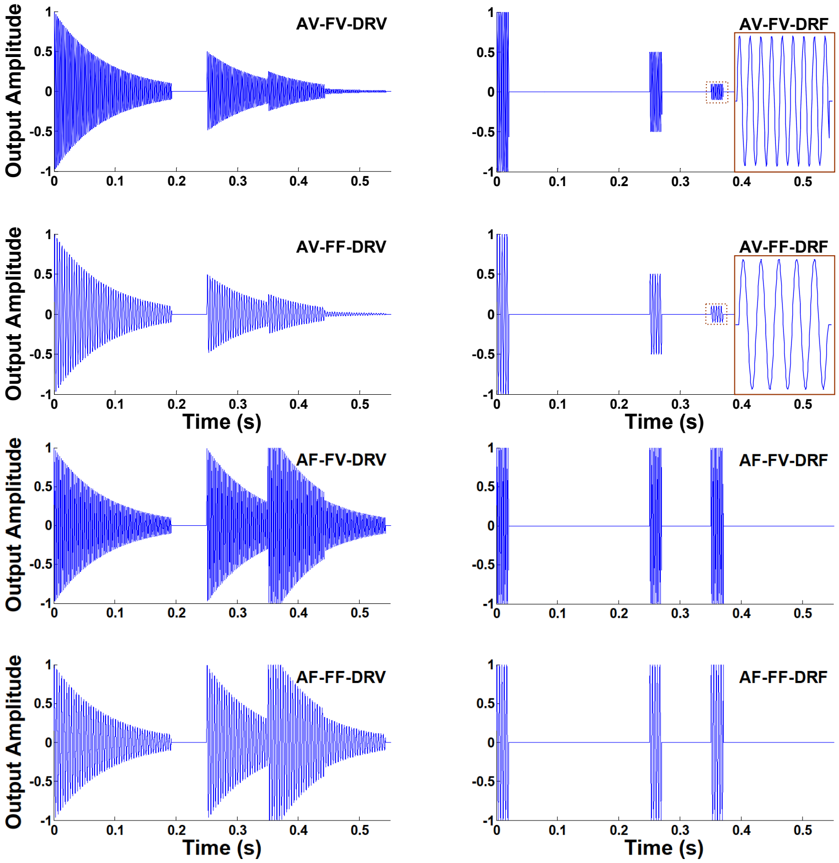


Fig. 5.12 Eight exemplar vibration waveforms from User Study 2. The waveforms were generated by three collisions with different impulses between a steel can and a steel bounding box. In the notation Ax-Fy-DRz, x, y, and z can be either V (varied) or F (fixed). A=V means that the vibration amplitude is modulated using the physics engine, and A=F means that the vibration amplitude is constant (1.5 g). If F=V, the vibration frequency is the natural frequency of the contacted object with a haptic focus. When F=F, the frequency is always 250 Hz. Likewise, the vibration is decayed using the decay rate if DR=V, but a constant envelope is used when DR=F (duration 20 ms).

Procedure

On each experimental trial, participants used one phone that ran the simulation application to perceive vibrotactile feedback to dynamic collision events and the other phone to enter their responses for the subjective rating of their experiences. The simulation program (Fig.

5.10) provided five buttons: three numbers ‘1’, ‘2’, and ‘3’, ‘PAUSE’, and ‘OK’. The three object pairs were randomly assigned to the three number buttons. Participants chose which object pair to interact with using the number buttons. They were instructed to use the two interaction methods in the equal proportion and experience each object pair for longer than 10 s. The latter was forced by deactivating/activating the ‘PAUSE’ and ‘OK’ buttons. Vibrotactile feedback was given at collisions using the rendering method designated to the trial. Participants wore headphones playing pink noise to block the sounds made by the vibration actuator.

After interacting with all the three object pairs, participants clicked the ‘PAUSE’ button, and the simulation application stopped vibration generation. Then participants rated the realism, harmony, and liking of the vibrotactile sensations using a UI displayed on the second phone (Fig. 5.11). The meanings of the three criteria given to participants were as follows: *Realism*—The vibrotactile feedback felt realistic; *Harmony*—The vibrotactile feedback was harmonious to the collisions between objects and walls; and *Liking*—I liked the vibrotactile feedback. Each item was rated using a seekbar shown on the screen. The left and right ends of the seekbars were labeled “Strongly Disagree” and “Strongly Agree,” respectively. The cursors of the three seekbars were initially positioned at the center. After entering responses using this UI, participants clicked the ‘OK’ button in the simulation program, and the next trial began. Participants’ responses on the seekbars were linearly converted to a 0–100 scale (50 neutral).

The experiment consisted of a training session and a main session. In the first block of a training session, participants experienced the eight experimental conditions in a random order. After two minutes of rest, the second block of training was repeated to help participants stabilize their rating criteria. In the main session, the eight experimental conditions were repeated ten times each, resulting in a total of 80 trials. The order of the experimental conditions was randomized per participant. Participants took a two-minute rest in every 20 trials. On average, 105 min was required to complete the experiment. Lastly, participants wrote down comments about the rating criteria they used.

5.3.2 Results and Discussion

Experimental results are presented in Fig. 5.13 and 5.14, which respectively show the mean scores for the three independent factors (amplitude, frequency, and decay rate) and those of the eight experimental conditions. For statistical analysis, three-way ANOVA was applied on the three main factors. Amplitude had statistically significant effects on realism and liking ($F(1,19) = 13.8, p = 0.0147$; $F(1,19) = 10.87, p = 0.0038$). Frequency had significant effects on all the three measures ($F(1,19) = 56.49, p < 0.001$; $F(1,19) = 32.98, p < 0.001$; $F(1,19) = 59.74, p < 0.001$). Decay rate had a significant effect on only realism ($F(1,19) = 6.30, p = 0.021$). Varying amplitude, frequency, or decay rate using PhysVib resulted in higher scores than any of its fixed counterpart. The score improvement averaged over the three measures was the greatest (23.2) with frequency, followed by amplitude (8.0) and decay rate (2.0).

The mean scores of AV-FV-DRV—76.4 for realism, 68.9 for harmony, and 71.2 for liking—can be regarded quite high given the general regression bias[3] in perceptual judgments [32]. This is in comparison to the much lower scores of AF-FF-DRV (realism 37.8, harmony 43.5, and liking 36.0) similar to the method of [104] and of AF-FF-DRF (realism 36.1, harmony 42.2, and liking 36.1) similar to the stimuli used in [85]. Recall that the two prior studies did not aim at physically plausible vibrotactile feedback for dynamic objects, but rather at the notification of abstract messages using a metaphor to physics.

Among the three independent factors, frequency seems to be the most important for providing plausible vibrotactile feedback given the greatest differences it caused in the three subjective scores. In the experiment, the natural frequency of the three object pairs varied in a wide range (80, 349, and 430 Hz), and this could have contributed to the greatest effect of frequency. Varying amplitude in response to collision impulse also had an important but less effect. Last, using the exponentially-decaying envelope had the least positive

[3] People are reluctant to give extreme answers. In some examples of the haptics research that used similar rating methods in the same scale (0–100), participants gave the similarity score of 82 for the same two real soft objects with homogeneous viscoelasticity [46] and 79 for the same two real inhomogeneous soft objects [107]. The median realism rating between real and virtual textures rendered using vibration only ranged from 50 to 90 [80].

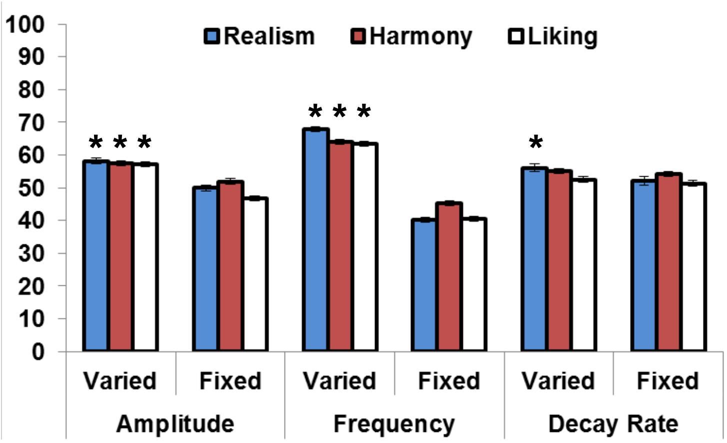


Fig. 5.13 Mean scores for the three independent factors. Error bars represent standard errors. An asterisk indicates a statistically significant difference between the two factor levels (varied and fixed).

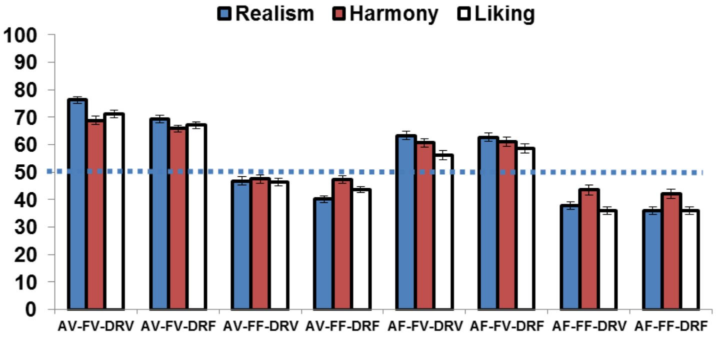


Fig. 5.14 Mean scores of the eight vibrotactile rendering methods. Error bars indicate standard errors.

effect. This implies that depending on the computing power available, one may simplify the costly exponential envelope to a linearly decreasing envelope, or even a rectangular envelope (pulse), without sacrificing the perceptual quality greatly.

Many participants commented that they assessed realism mainly based on different sensations between different objects, as well as residual vibration. They rated harmony high if vibration amplitude and onset time were in good agreement with the visual collisions.

In summary, the results of User Study 2 indicate that PhysVib can offer physically plausible vibrotactile feedback that is harmonious with the visually-perceived physical properties of colliding objects. The subjective scores of PhysVib were sufficiently high, and the scores were by far greater than those of the two previous methods [85, 104]. The results also classified amplitude and frequency to be more crucial parameters for physically plausible vibrotactile feedback than decay rate.

5.4 Conclusions

This section described the design and implementation of PhysVib, a software library on a mobile platform that delivers physically plausible vibrotactile feedback for collision events. PhysVib is built upon a physic engine and the realty-based vibration model that uses empirically-confirmed exponentially-decaying sinusoidal functions. PhysVib consists of three main components—collision catcher, vibration manager, and vibration converter—in a multi-rate rendering architecture seamlessly integrated with the operation of a physics engine. The collision catcher communicates with a physics engine and computes the variables required for vibrotactile feedback. The vibration manager makes normalized vibration signals using the reality-based model. The vibration converter transforms the normalized signals to the commands tailored to the actuator. PhysVib requires application developers to supply only two additional material parameters for each graphical object while taking care of the rest, leading to greater productivity in the development of vibration feedback-enabled applications.

The performance of PhysVib was estimated in the two user studies. User Study 1 demonstrated that the reality-based model is better or at least comparable to sound-to-vibration synthesis, a physically more accurate but computationally far less desirable alternative, in terms of harmony to visually-observed collision events. User Study 2 showed that PhysVib can provide high realism and harmony to users along with high user liking and it outperforms the two simpler prior approaches.

To the best of our knowledge, PhysVib is the first of its kind, and there can be many directions for future extension. Also, PhysVib showed its plausible perceptual quality to collision events through two user studies. But this library was designed to superimpose all vibration waveforms to a single normalized signal so spatial information were not able to be transferred to users although PhysVib could calculate them. Moreover, the haptic enchanters are the great haptic modules that can be utilized as the secondary information channel to transfer 2D spatial information as in Chapter4.1. In the next chapter, an improvement of PhysVib to distribute the vibrations for spatial information and an integration of the improved PhysVib and the haptic enchanters are described.

Chapter 6

Multi-Actuator Extension of PhysVib and its Integration with Haptic Enchanters

Many researchers have been tried to develop vibrotactile feedback authoring tools to reduce the feedback design cost and improve the stimuli quality as described in Section 2.3. Some of them supported the multiple-actuator authoring [82, 60, 57, 73] and corresponding signals could be exported to a file or directly rendered as vibrotactile signals. This approach is appropriate for a general feedback design because it has no context between vibrotactile signals, vibration systems, and target applications.

In this thesis, haptic enchanters were able to provide a 6 by 5 grid information using 2D phantom sensations as in Section 4.2. Also, PhysVib showed its perceptual realism and liking in the experiment application in Chapter 5. This PhysVib was designed to receive all geometric information from collisions in a simulation that is essential to apply 1D or 2D phantom sensations. By integrating these, PhysVib has an potential to provide its physically plausible vibrations to multiple haptic enchanters by utilizing 2D phantom sensations as in Section 4.2.

6.1 Structure of Extended PhysVib

PhysVib was implemented to render a single vibration signal from multiple vibration sources, haptically focused objects, using collision catcher, vibration manager, and vibration converter. In an application using PhysVib, a user was assumed to touch all haptically focused objects *simultaneously* regardless the number of objects. Physically, a single actuator was regarded as a connector between a user in the real world and focused objects in a simulation.

A main goal of the extension of PhysVib by using haptic enchanters is providing physically plausible vibrations where the vibrations occur, in other words, physically plausible spatial vibrotactile feedback. To implement the main goal, a definition of using scenario must be preceded to confirm how the structure of PhysVib needs to be changed. Before going further, this extension considered only two configurations of PHONE and RING and limited the number of haptic enchanters to one, two, or four. These limitations were made because only their perceptual qualities were known from this thesis while other conditions were unveiled yet.

A using scenario of using multiple haptic enchanters is as follows. If a user uses multiple haptic enchanters to feel physically plausible spatial vibrotactile feedback from collisions, he or she follows these two steps: a system configuration and an interaction. In the system configuration, a user puts any number of haptic enchanters on where he or she wants to attach to. In the interaction step, the user controls objects of a PhysVib application and each haptic enchanter renders physically plausible spatial vibrations by using phantom sensations. The user can reconfigure the haptic enchanters at anytime during the interaction step.

From the using scenario, main requirements for PhysVib extension are twofold: information of a haptic enchanter configuration and extension to multiple vibration signal channels. PhysVib needs to know where haptic enchanters are attached to and render vibrotactile signals for each haptic enchanter.

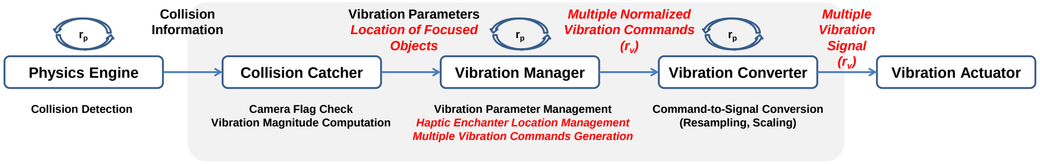

Collision Information
Vibration Parameters
Location of Focused Objects
Multiple Normalized Vibration Commands (r_v)
Multiple Vibration Signal (r_v)
r_p
r_p
r_p
Physics Engine
Collision Catcher
Vibration Manager
Vibration Converter
Vibration Actuator
Collision Detection
Camera Flag Check
Vibration Magnitude Computation
Vibration Parameter Management
Haptic Enchanter Location Management
Multiple Vibration Commands Generation
Command-to-Signal Conversion
(Resampling, Scaling)

Fig. 6.1 Extended structure and internal processes of PhysVib. Changed or added features compared to Figure 5.2 are colored in red.

6.2 Processes

Extended PhysVib shares the common structure with the original PhysVib, but some details are changed to support multiple haptic enchanters (Fig. 6.1). The parallel processes of vibration manager and vibration converter are not changed, but vibration manager now embeds the the number of haptic enchanters (1, 2, or 4) and their vector locations.

The process from a collision detection to a computation of vibration parameters still remains the same. However, collision catcher sends locations of focused objects in addition to the vibration parameters to vibration manager when a collision occurs. Now the role of vibration manager is to make vibration signals from the transferred parameters and distribute them to each haptic enchanter using phantom sensations with a high sampling rate $r_v(r_v \gg r_p)$. Also, the distributed signal to each enchanter is normalized and vibration manager sends segments of the normalized vibration signal to vibration converter. The role of vibration converter is identical to that of the original PhysVib except using multiple channels.

In the follow subsections, changed features of collision catcher, vibration manager, and vibration converter are described in detail. Features that have not been changed are briefly explained to help understanding.

6.2.1 Collision Catcher

Features of collision catcher in the original PhysVib were threefold: calculation of vibration parameters, normalization of vibration amplitudes, and transfer of vibration parameters. For extended PhysVib, former two features are already implemented sufficiently and do not need any modification. In the vibration parameter transfer, collision catcher makes a linked list L^E_{col} in which each node contains f, τ, A, and $\mathbf{d}_f$ for a collision. The list L^E_{col} has one more element of $\mathbf{d}_f$ than L_{col} which is a location vector of haptically focused fixture. L^E_{col} is then passed to the vibration manager for vibration command generation.

6.2.2 Vibration Manager

In Chapter 5.1.3, vibration manager of original PhysVib 1) stores parameters from L_{col}, 2) computes a vibration waveform X for $(1/r_p)$s for each node in L_{vib}, 3) superimposes Xs and normalizes it with the amplitude bound of $[-1, 1]$, and 4) sends the superimposed vibration signal to vibration converter. In the extended PhysVib, 1), 2), and 4) are changed and vibration manager updates and keeps the number of haptic enchanters n_e and location vectors $\mathbf{d}_{ej}$ to support multiple haptic enchanters.

First, vibration manager maintains a linked list L^E_{vib} to store parameters for vibration generation. Whenever L^E_{col} is passed from collision catcher, a node is created in L^E_{vib} for each new vibration parameter node in L^E_{col}, and f, τ, A, and $\mathbf{d}_f$ are copied into it from the corresponding node of L^E_{col}. Additional parameters of t_0 and t_d are also added as the original vibration manager.

Second, a vibration waveform X_j of j-th haptic enchanter for $(1/r_p)$s for each node in L^E_{vib} is computed

$$X_j = \left\{ A_j x \left(t + \frac{1}{r_v} i \right) \middle| \, 0 \le i < \left\lceil \frac{r_v}{r_p} \right\rceil \text{ for integer } i \right\}, \tag{6.1}$$

where A_j is a coefficient calculated using phantom sensations as

$$A_j = A \left(1 - \frac{|d^x_f - d^x_{ej}|}{D^x} \right)^{\gamma} \left(1 - \frac{|d^y_f - d^y_{ej}|}{D^y} \right)^{\gamma}, \tag{6.2}$$

where A is the passed amplitude from the collision catcher, D^x, D^y are the maximum distance (default values are of a screen resolution) in x and y axis, d^x_f, d^y_f are the x and y coordinate of the focused fixture, and d^x_{ej}, d^y_{ej} are the x and y coordinate of j-th haptic enchanter. Using a 1 to the γ renders phantom sensations as applying the linear rendering method as in Section 4.1. All X_j waveforms follow the step 3) to generate j-th superimposed vibration signal.

At the last step, vibration manager sends the n_e superimposed vibration signals to vibration converter as a $\left\lceil \frac{r_v}{r_p} \right\rceil$ by n_e array.

6.2.3 Vibration Converter

The role of vibration converter is still transforming the normalized vibration command to actual vibration signal for a target actuator system. In the extended PhysVib, the target system is haptic enchanters and each superimposed vibration signal corresponds to each haptic enchanter.

6.3 Implementation

For rendering two haptic enchanters, using the stereo audio channel was sufficient to render 1D phantom sensations.

Currently, there exists no system that can transfer three or more than three continuous waveforms simultaneously without loss. As a substitute system, the vibration manager and the vibration converter were implemented in an embedded system to actuate up to four actuators. The embedded system was extended from the hardware used in user studies in Chapter 3 and 4.

It is identical that receiving L^E_{col}, however, the vibration manager just stacks nodes in L^E_{vib} with A_j. For this process, the embedded system has four stacks for each haptic enchanters and each stack receives a vibration command node with its modulated amplitude A_j. A rendering process runs at 100 Hz in the embedded system to calculate X_j for each haptic enchanter by calculating each stacked nodes. However, the maximum number of vibration commands in the system was only eight because of the lack of computing power and the RISC architecture requiring multiple clocks for a floating point calculation (66 MHz).

Chapter 7

Conclusions

This thesis asserts the contribution of attachable and detachable haptic modules (haptic enchanters) through three perceptual experiments and an automated feedback authoring library. Lastly, the library is integrated to the haptic modules.

First, the author investigated how to develop haptic enchanters and implemented prototypes for further experiments. Because the required expertise to implement a neat small module integrating electronics and a power was too high, prototypes are developed to see potential advantages as a platform with the ideal size. Too vast using scenarios forced to select attaching to a mobile phone (PHONE) and human fingers (RING) as representative configurations, and a silicone adhesive pad of box-type and a ring-type attachment methods correspond to them. PHONE requires a vibration attenuation layer between an actuator and a rigid phone, or single or multiple enchanters do not make difference due to the vibration propagation. Through an extensive measurement using 9,000 conditions, a silicone layer of shore hardness of 0030, softner ratio of 15 %, and thickness of 3 mm of thickness is found and shows over 90% of vibration attenuation.

Second, perceptual qualities using haptic enchanters are estimated through three user studies. As a performance measure of a platform for information transmission, information transfer (IT) is used in User Study 1 and 2. In the representative configurations of PHONE and RING, spatiotemporal patterns without illusory sensations provided the high information capacity as a main information channel. Stationary spatial patterns using illusory effect showed low IT values for transferring exact spatial cues, however, the results supported some potential to a secondary information channel. We redesigned the second user study to measure more detailed user responses, and this system could provide at least 5 by 4 grid spatial cue to a user. Through all these user studies, both of PHONE and RING showed the similar extent of these perceptual qualities.

The author also developed PhysVib: a software library that generates physically plausible vibrotactile feedback to collision events. PhysVib only requires a natural frequency and a decay rate extracted from a recorded sound from the real collision. This extraction method is quite simple and does not need in-depth understanding of haptics. Although these low requirements, PhysVib could provide perceptually realistic, liking, and harmonious vibrotactile feedback comparing to similar methods used in the literature. This library then extended to support multiple haptic enchanters by adopting rendering methods of phantom sensations.

요 약 문

탈부착이 가능한 진동 피드백 장치와 그 응용

최근 많은 장치들을 활용한 상호작용이 빈번히 일어나면서, 사용자들은 다감각 표시 장치를 이용해 많은 정보를 획득하고 있다. 햅틱 인터페이스를 활용한 수많은 연구가 있음에도 불구하고, 다감각 표시 장치에 햅틱 하드웨어를 내장하는 데 드는 높은 비용 때문에 거의 상용화되고 있지는 못한 상황이다. 이 연구의 목적은 탈부착이 가능한 햅틱 모듈인 햅틱 인챈터를 사용하여 사용자가 원하는 곳에 높은 정보 전달 (IT)이 가능하게 하는 햅틱 인터페이스를 만드는 것이다.

햅틱 인챈터는 진동자, 부착부, 그리고 추가적인 진동 고립부로 이루어져 있다. 총 아홉 종류의 상자 타입과 반지 타입의 프로토타입이 개발되었고, 이 프로토타입은 단단하고 평평한 표면을 가진 모바일 폰이나 손가락에 부착하여 진동을 주도록 개발되었으며, 이 중 세 종류에는 진동 고립부가 제외되었다. 진동 고립부는 9000가지의 실험 조건 및 10번의 반복 측정을 통해 결정되었으며, 측정 결과 쇼어 경도계 0030의 기본 실리콘에 연화제를 15 %만큼 섞은 실리콘 혼합물을 3 mm의 두께로 사용하는 경우 손으로 가볍게 쥔 상태에서도 90 퍼센트 이상의 진동 감쇄율을 보였다. 반지 타입의 프로토타입은 사람의 손가락에 직접적으로 부착되므로 진동 고립부를 부착할 필요가 없었다.

햅틱 인챈터의 인지적인 효과는 챕터 3과 4의 세 종류의 사용자 실험으로 측정되었다. 한 플랫폼의 정보 전달력을 측정하는 값인 정보 전달 (IT)가 챕터 3과 4에 있는 두 종류의 사용자 실험에서 측정되었다. 대표적인 사용 환경인 PHONE과 RING에서,

2개, 3개, 4개를 사용한 시공간 진동 패턴은 높은 진동 전달 능력을 보여주었다(4.55-7.06 bits). 정적인 공간 패턴에 대해서는, 네 개의 햅틱 인챈터를 사용한 2차원 환상 감각을 이용하는 경우 3 by 3, 4 by 4, 5 by 5의 2차원 가상 격자 공간에 대해 1.89-2.54 bits의 정보 전달 값을 보였다. 이 결과는 2D 환상 감각이 주 정보 채널이 아닌 보조 정보 채널로서의 가능성을 가지고 있음을 보여주었기 때문에, 환상 감각의 해상도가 좀 더 정확히 측정되어야 할 필요가 있었다. 측정 결과 선형 렌더링과 로그 렌더링을 이용한 PHONE에 대해서는 5 by 5, 5 by 4의 인지 해상도를 표현할 수 있음을 보였으며, 선형 렌더링과 로그 렌더링을 이용한 RING에서는 각각 6 by 5, 7 by 4 의 인지 해상도를 보일 수 있었다. 이 결과들은 햅틱 인챈터가 효율적이면서도 편리한 통신 보조기구로 쓰일 수 있음을 보여준다.

이러한 햅틱 인챈터들을 활용하기 위한 응용 프로그램으로 패턴 저작 도구를 제작하였다. PhysVib는 모바일 플랫폼에 사용되어 물리 엔진의 충돌 이벤트에 대해 진동을 자동적으로 생성시켜주는 소프트웨어 라이브러리로 개발되었다. PhysVib는 물리 엔진과 동시에 구동되며, 낮은 갱신률로 충돌 이벤트를 감지하며 높은 갱신률로 지수 감쇄 정현파 모델을 사용한 진동 피드백을 생성한다. 사용자 실험에서는 이 모델이 실제 충돌 소리를 사용한 진동보다 시각적 이벤트와 더 잘 어울림을 보였다. 또 다른 실험에서는 PhysVib에 위 모델을 포함한 여덟 종류의 진동 생성 모델을 적용하여 그 인지적 성능을 비교하였다. 실험 결과, PhysVib에서 사용한 지수 감쇄 정현파 모델이 시각적 이벤트와의 어울림, 사실성, 그리고 호감에 대해 가장 좋은 결과를 보였다. PhysVib는 사실적인 진동 패턴을 제공해주는 효율적인 라이브러리이며, 진동 패턴 개발 시간을 획기적으로 줄여줄 수 있다. 여러 개의 햅틱 인챈터 역시 PhysVib에 2D 환상 감각 렌더링 방법을 적용하는 것으로 사용 가능하다.

REFERENCES

[1] D. S. Alles. Information transmission by phantom sensations. *IEEE Transactions on Man-Machine Systems*, MMS-11(1):85–91, 1970.

[2] O. R. Astley and V. Hayward. Multirate haptic simulation achieved by coupling finite element meshes through Norton equivalents. In *Proc. IEEE ICRA*, pages 989–994, 1998.

[3] F. Avanzini and P. Crosato. Integrating physically based sound models in a multimodal rendering architecture. *Comput. Animat. Virtual Worlds*, 17(3-4):411–419, 2006.

[4] F. Avanzini and D. Rocchesso. Modeling collision sounds: Non-linear contact force. In *Proc. DAFx-01*, pages 61–66, 2001.

[5] M. Azadi and L. Jones. Identification of vibrotactile patterns: Building blocks for tactons. In *Proceedings of the IEEE World Haptics Conference (WHC)*, pages 347–352, 2013.

[6] P. Bach-Y-Rita, C. C. Collins, F. A. Saunders, B. White, and L. Scadden. Vision substitution by tactile image projection. *Nature*, 1969.

[7] A. Barghout, J. Cha, A. E. Saddik, J. Kammerl, and E. Steinbach. Spatial resolution of vibrotactile perception on the human forearm when exploiting funneling illusion. In *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, pages 19–23, 2009.

[8] J. C. Bliss, M. H. Katcher, C. H. Rogers, and R. P. Shepard. Optical-to-tactile image conversion for the blind. *Man-Machine Systems, IEEE Transactions on*, 11(1):58–65, 1970.

[9] S. J. Bolanowski, Jr., G. A. Gesheider, R. T. Verrillo, and C. M. Checkosky. Four channels mediate the mechanical aspects of touch. *Journal of Acoustical Society of America*, 84(5):1680–1694, 1988.

[10] S. Brewster and L. M. Brown. Tactons: Structured tactile messages for non-visual information display. In *Proceedings of the 5th Australasian User Interface Conference*, pages 15–23, 2004.

[11] P. L. Brooks and B. J. Frost. The development and evaluation of a tactile vocoder for the profoundly deaf. *Canadian Journal of Public Health*, 77:108–113, 1986.

[12] L. M. Brown, S. A. Brewster, and H. C. Purchase. Multidimensional tactons for non-visual information presentation in mobile device. In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 231–238. ACM, 2006.

[13] L. M. Brown and T. Kaaresoja. Feel who's talking: Using tactons for mobile phone alerts. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 604–609. ACM, 2006.

[14] M. C. Çavuşoğlu and F. Tendick. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. pages 2458–2465, 2000.

[15] E. Catto. *Box2D v2.3.0 User Manuel*.

[16] L. H. Chan and K. S. Choi. Integrating PhysX and OpenHaptics: Efficient force feedback generation using physics engine and haptic devices. In *Proc. JCPC*, pages 853–858. IEEE, 2009.

[17] A. Chang and C. O'Sullivan. Audio-haptic feedback in mobile phones. In *Proceeding of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1264–1267, 2005.

[18] H.-Y. Chen, J. Park, S. Dai, and H. Z. Tan. Design and evaluation of identifiable key-click signals for mobile devices. *IEEE Transactions on Haptics*, 4(4):229–241, 2011.

[19] H.-Y. Chen, J. Santos, M. Graves, K. Kim, and H. Z. Tan. Tactor localization at the wrist. *(EuroHaptics 2008)*, 5024:209–218, 2008.

[20] S. Choi and K. J. Kuchenbecker. Vibrotactile display: Perception, technology, and applications. *Proc. IEEE*, 101(9):2093–2104, 2013.

[21] S. A. Cholewiak, K. Kim, H. Z. Tan, and B. D. Adelstein. A frequency-domain analysis of haptic gratings. *IEEE Trans. Haptics*, 3(1):3–14, 2010.

[22] C. C. Collins. Tactile television-mechanical and electrical image projection. *Man-Machine Systems, IEEE Transactions on*, 11(1):65–71, 1970.

[23] P. R. Cook. Physically informed sonic modeling (phism): Synthesis of percussive sounds. *Computer Music Journal*, pages 38–49, 1997.

[24] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. AK Peters, Ltd., 2007.

[25] J. C. Craig. *Pictorial and abstract cutaneous displays*. Psychonomic Society, 1974.

[26] F. Danieau, A. Lécuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie. Enhancing audiovisual experience with haptic feedback: A survey on HAV. *IEEE Transtions Haptics*, 6(2):193–205, 2013.

[27] D. DiFilippo and D. K. Pai. The AHI: An audio and haptic interface for contact interactions. In *Proc. ACM UIST*, pages 149–158, 2000.

[28] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamics Systems*. Addison-Wesley Publishing Company, Reading, MA, U.S.A., second edition, 1990.

[29] A. Gallace, H. Z. Tan, and C. Spence. The body surface as a communication system: The state of the art after 50 years. *Presence*, 16(6):655–676, 2007.

[30] R. H. Gault. Progress in experiments on tactual interpretation of oral speech. *Journal of Abnormal Psychology and Social Psychology*, 19(2):155–159, 1924.

[31] G. A. Gescheider. *Psychophysics: The Fundamentals*. Lawrence Erlbaum Associate, Mahwah, NJ, USA, 3rd edition, 1997.

[32] G. A. Gescheider. *Psychophysics: The Fundamentals*. Lawrence Erlbaum Associate, Mahwah, NJ, USA, 3rd edition, 1997.

[33] E. B. Goldstein. *Sensation and Perception*. Wadsworth-Thomson Learning, Pacific Grove, CA, USA, 6th edition, 2002.

[34] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.

[35] E. Hoggan and S. Brewster. New parameters for tacton design. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, pages 2417–2422, 2007.

[36] E. Hoggan, S. A. Brewster, and J. Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1573–1582. ACM, 2008.

[37] K. Hong, J. Lee, and S. Choi. Demonstration-based vibrotactile pattern authoring. In *Proceedings of the International Conference on Tangible, Embedded and Embodied Interaction (TEI)*, pages 219–222. ACM, 2013.

[38] C. M. R. W. M. R. Hong Z. Tan, Nathaniel I. Durlach. Information transmission with a multi-finger tactual display. *Perception & Psychophysics*, 61(6):993–1008, August 1999.

[39] I. Hwang and S. Choi. Improved haptic music player with auditory saliency estimation. *Lecture Notes in Computer Science (EuroHaptics 2014)*, 2014. To be presented.

[40] I. Hwang, H. Lee, and S. Choi. Real-time dual-band haptic music player for mobile devices. *IEEE Transactions on Haptics*, 3(3):340–351, 2013.

[41] I. Hwang, K. E. MacLean, M. Brehmer, J. Hendy, A. Sotirakopoulos, and S. Choi. The haptic crayola effect: Exploring the role of naming in learning haptic stimuli. In *Proceedings of the IEEE World Haptics Conference (WHC)*, pages 385–390. IEEE, 2011.

[42] I. Hwang, J. Seo, M. Kim, and S. Choi. Vibrotactile perceived intensity for mobile devices as a function of direction, amplitude, and frequency. *IEEE Transactions on Haptics*, 6(3):352–362, 2013.

[43] A. Israr and I. Poupyrev. Tactile brush: Drawing on skin with a tactile grid display. In *Proceedings of the SIGCHI Annual Conference on Human Factors in Computing Systems (CHI)*, pages 2019–2028. ACM, 2011.

[44] R. Jacob, P. Mooney, and A. C. Winstanley. Guided by touch: Tactile pedestrian navigation. In *Proceedings of the 1st International Workshop on Mobile Location-based Service (MLBS)*, pages 11–20. ACM, 2011.

[45] S. Jeon and S. Choi. Haptic augmented reality: Taxonomy and an example of stiffness modulation. *Presence: Teleoperators and Virtual Environments*, 18(5):387–408, 2009.

[46] S. Jeon, S. Choi, and M. Harders. Rendering virtual tumors in real tissue mock-ups using haptic augmented reality. *IEEE Trans. on Haptics*, 5(1):77–84, 2012.

[47] L. A. Jones and S. J. Lederman. *Human Hand Function*. Oxford University Press, 2006.

[48] T. Kaaresoja, E. Hoggan, and E. Anttila. Playing with tactile feedback latency in touchscreen interaction: Two approaches. In *Human-Computer Interaction–INTERACT 2011*, pages 554–571. Springer, 2011.

[49] J. Kang, J. Lee, H. Kim, K. Cho, S. Wang, and J. Ryu. Smooth vibrotactile flow generation using two piezoelectric actuators. *IEEE Transactions on Haptics*, 5(1):21–32, 2012.

[50] M. Kim, S. Lee, and S. Choi. Saliency-driven real-time video-to-tactile translation. *IEEE Transactions on Haptics*, 5, 2014. To appear (online published).

[51] S.-Y. Kim and J. C. Kim. Vibrotactile rendering for a traveling vibrotactile wave based on a haptic processor. *IEEE Transactions on Haptics*, 5(1):14–20, 2012.

[52] S.-Y. Kim and J. C. Kim. Vibrotactile rendering for a traveling vibrotactile wave based on a haptic processor. *IEEE Transactions on Haptics*, 5(1):14–20, 2012.

[53] Y. Kim, J. Lee, and G. J. Kim. Designing of 2d illusory tactile feedback for hand-held tablets. In *Lecture Notes on Computer Science*, volume LNCS 9299 (INTERACT 2015), pages 10–17. 2015.

[54] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):219–230, 2006.

[55] E. C. Lechelt. Temporal numerosity discrimination: Intermodal comparisons revisited. *British Journal of Psychology*, 66(1):101–108, 1975.

[56] I. Lee and S. Choi. Vibrotactile guidance for drumming learning: Method and perceptual assessment. In *Haptics Symposium (HAPTICS), 2014 IEEE*, pages 147–152. IEEE, 2014.

[57] J. Lee and S. Choi. Evaluation of vibrotactile pattern design using vibrotactile score. In *Proceedings of the IEEE Haptics Symposium*, pages 231–238, 2012.

[58] J. Lee and S. Choi. Real-time perception-level translation from audio signals to vibrotactile effects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2567–2576. ACM, 2013.

[59] J. Lee, Y. Kim, and G. J. Kim. Funneling and saltation effects for tactile interaction with virtual objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 3141–3148. ACM, 2012.

[60] J. Lee, J. Ryu, and S. Choi. Vibrotactile score: A score metaphor for designing vibrotactile patterns. In *Proceedings of the World Haptics Conference*, pages 302–307. IEEE, 2009.

[61] S. C. Lee and T. Starner. Buzzwear: Alert perception in wearable tactile displays on the wrist. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 433–442, 2010.

[62] R. W. Lindeman, R. Page, Y. Yanagida, and J. L. Sibert. Towards full-body haptic feedback: The design and deployment of a spatialized vibrotactile feedback system. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, 146–149, 2004.

[63] J. G. Linvill and J. C. Bliss. A direct translation reading aid for the blind. *Proceedings of the IEEE*, 54(1):40–51, 1966.

[64] J. Liu and H. Ando. Hearing how you touch: real-time synthesis of contact sounds for multisensory interaction. In *Human System Interactions, 2008 Conference on*, pages 275–280. IEEE, 2008.

[65] J. Luk, J. Pasquero, S. Little, K. MacLean, V. Levesque, and V. Hayward. A role for haptics in mobile interaction: Initial design using a handheld tactile display prototype. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 171–180. ACM, 2006.

[66] K. E. MacLean. Foundations of transparency in tactile information design. *IEEE Transactions on Haptics*, 1(2):84–95, 2008.

[67] K. E. MacLean and M. Enriquez. Perceptual design of haptic icons. In *Proceedings of Eurohaptics*, pages 351–363, 2003.

[68] M. Matscheko, A. Ferscha, A. Riener, and M. Lehner. Tactor placement in wrist worn wearables. In *Proceedings of the IEEE International Sympsoium on Wearable Computing (ISWC)*, pages 1–8, 2010.

[69] S. Nanayakkara, E. Taylor, L. Wyse, and S. H. Ong. An enhanced musical experience for the deaf: Design and evaluation of a music display and a haptic chair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 337–346. ACM, 2009.

[70] H. Nicolau, J. Guerreiro, T. Guerreiro, and L. Carriço. Ubibraille: Designing and evaluating a vibrotactile braille-reading device. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 23. ACM, 2013.

[71] H. Nicolau, K. Montague, T. Guerreiro, A. Rodrigues, and V. L. Hanson. Holibraille: Multipoint vibrotactile feedback on mobile devices. In *Proceedings of the 12th Web for All Conference*, page 30. ACM, 2015.

[72] A. M. Okamura, M. R. Cutkosky, and J. T. Dennerlein. Reality-based models for vibration feedback in virtual environments. *IEEE/ASME Trans. Mechatronics*, 6(3):245–252, 2001.

[73] S. Panëels, M. Anastassova, and L. Brunet. TactiPEd: Easy prototyping of tactile patterns. *Lecture Notes in Computer Science (INTERACT 2013, Part II)*, LNCS 8118:228–245, 2013.

[74] S. Paneels and J. C. Roberts. Review of designs for haptic data visualization. *IEEE Transactions on Haptics*, 3(2):119–137, 2010.

[75] G. Park, S. Choi, K. Hwang, S. Kim, J. Sa, and M. Joung. Tactile effect design and evaluation for virtual buttons on a mobile device touchscreen. In *Proc. MobileHCI*, pages 11–20. ACM, 2011.

[76] J. Pasquero, J. Luk, S. Little, and K. Maclean. Perceptual analysis of haptic icons: An investigation into the validity of cluster sorted mds. In *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 437–444, 2006.

[77] J. Pasquero, S. J. Stobbe, and N. Stonehouse. A haptic wristwatch for eyes-free interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 3257–3266. ACM, 2011.

[78] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. *Comput. Graph. Forum*, 21(3):279–287, 2002.

[79] D. Rocchesso and F. Fontana. *The sounding object*. Mondo estremo, 2003.

[80] J. M. Romano and K. J. Kuchenbecker. Creating realistic virtual textures from contact acceleration data. *IEEE Transactions on Haptics*, 5(2):109–119, 2012.

[81] D. Ryu, G.-H. Yang, and S. Kang. T-Hive: Vibrotactile interface presenting spatial information on handle surface. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 683–688. IEEE, 2009.

[82] J. Ryu and S. Choi. posVibEditor: Graphical authoring tool of vibrotactile patterns. In *Proceedings of the IEEE International Workshop on Haptic, Audio and Visual Environments and Games*, pages 120–125, 2008.

[83] J. Ryu, J. Jung, G. Park, and S. Choi. Psychophysical model for vibrotactile rendering in mobile devices. *Presence: Teleoperators and Virtual Environments*, 19(4):1–24, 2010.

[84] J. Ryu, C.-W. Lee, and S. Choi. Improving vibrotactile pattern identification for mobile devices using perceptually transparent rendering. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 257–260. ACM, 2010.

[85] Y. Sekiguchi, K. Hirota, and M. Hirose. Haptic interface using estimation of box contents metaphor. *ICAT2003*, pages 197–202, 2003.

[86] J. Seo and S. Choi. Initial study of creating linearly moving vibrotactile sensation on mobile device. In *Proceedings of the Haptics Symposium (HS)*, pages 67–70. IEEE, 2010.

[87] J. Seo and S. Choi. Perceptual analysis of vibrotactile flows on a mobile device. *IEEE Transactions on Haptics*, 6(4):522–527, 2013.

[88] J. Seo and S. Choi. Edge flows: Improving information transmission in mobile devices using two-dimensional vibrotactile flows. In *Proceedings of the IEEE World Haptics Conference (WHC)*, pages 25–30, 2015.

[89] S. Shin, R. H. Osgouei, K.-D. Kim, and S. Choi. Data-driven modeling of isotropic haptic textures using frequency-decomposed neural networks. In *Proceedings of the IEEE World Haptics Conference (WHC)*, pages 131–138, 2015.

[90] K. O. Sofia and L. Jones. Mechanical and psychophysical studies of surface wave propagation during vibrotactile stimulation. *Haptics, IEEE Transactions on*, 6(3):320–329, 2013.

[91] J. Sreng, F. Bergez, J. Legarrec, A. Lécuyer, and C. Andriot. Using an event-based approach to improve the multimodal rendering of 6DOF virtual contact. In *Proc. ACM VRST*, pages 165–173, 2007.

[92] M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computer & Graphics*, 21(4):393–404, 1997.

[93] I. R. Summers. *Tactile aids for the hearing impaired.* Whurr Publishers London, 1992.

[94] C. Swindells, E. Maksakov, K. E. Maclean, and V. Chung. The role of prototyping tools for haptic behavior design. In *Proc. HAPTICS*, pages 161–168. IEEE, 2006.

[95] S. Tachi, K. Tanie, K. Komiyama, K. Asaba, Y. Tomita, and M. Abe. Information transmission by two-dimensional electrocutaneous phantom sensation. In *Proceedings of the 8th Annual Northeast Bioengineering Conference*, pages 258–262, 1980.

[96] D. Tam, K. E. MacLean, J. McGrenere, and K. J. Kuchenbecker. The design and field observation of a haptic notification system for timing awareness during oral presentations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1689–1698. ACM, 2013.

[97] H. Z. Tan, N. I. Durlach, C. M. Reed, and W. M. Rabinowitz. Information transmission with a multifinger tactual display. *Perception & Psychophysics*, 61(6):993 – 1008, 1999.

[98] H. Z. Tan, C. M. Reed, and N. I. Durlach. Optimum information-transfer rates for communication through haptic and other sensor modalities. *IEEE Transactions on Haptics*, 3(2):98–108, 2010.

[99] D. Ternes and K. E. MacLean. Designing large sets of haptic icons with rhythm. *Lecture Notes on Computer Science (EuroHaptics 2008)*, 5024:199–208, 2008.

[100] K. van den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 537–544, New York, NY, USA, 2001. ACM.

[101] J. B. F. van Erp, H. A. H. C. van Veen, and C. Jansen. Waypoint navigation with a vibrotactile waist belt. *ACM Transactions on Applied Perception*, 2(2):106–117, 2005.

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

[103] B. W. White, F. A. Saunders, L. Scadden, P. Bach-Y-Rita, and C. C. Collins. Seeing with the skin. *Perception & Psychophysics*, 7(1):23–27, 1970.

[104] J. Williamson, R. Murray-Smith, and S. Hughes. Shoogle: excitatory multimodal interaction on mobile devices. In *Proc. ACM CHI*, pages 121–124, 2007.

[105] G.-H. Yang, M. sub Jin, Y. Jin, and S. Kang. T-mobile: Vibrotactile display pad with spatial and directional information for hand-held device. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5245–5250. IEEE, 2010.

[106] K. Yatani and K. N. Truong. Semfeel: A user interface with semantic tactile feedback for mobile touch-screen devices. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 111–120. ACM, 2009.

[107] S. Yim, S. Jeon, and S. Choi. Data-driven haptic modeling and rendering of viscoelastic and frictional responses of deformable objects. *IEEE Trans. on Haptics*, 2016. To appear.

[108] Y. Yoo, I. Hwang, and S. Choi. Consonance of vibrotactile chords. *IEEE Trans. Haptics*, 7(1):3–13, 2014.

Publications

Publications

International Journals

1. Gunhyuk Park and Seungmoon Choi, “A Physics-Based Vibrotactile Feedback Library for Collision Events”,*IEEE Transactions on Haptics*, Accepted for publication.

2. Jaemin Chun, In Lee, Gunhyuk Park, Jongman Seo, Seungmoon Choi, and Sung H. Han, “Efficacy of Haptic Blind Spot Warnings Applied through a Steering Wheel or a Seatbelt”,*Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 21, pp. 231-241, 2013.

3. Jaemin Chun, Sung H. Han, Gunhyuk Park, Jongman Seo, In Lee, and Seungmoon Choi, “Evaluation of Vibrotactile Feedback for Forward Collision Warning on the Steering Wheel and Seatbelt,” *International Journal of Industrial Ergonomics*, vol. 42, no. 5, pp. 443-448, 2012.

4. Jonghyun Ryu, Jaemin Chun, Gunhyuk Park, Seungmoon Choi, and Sung H. Han, “Vibrotactile Feedback for Information Delivery in the Vehicle”, *IEEE Transactions on Haptics*, vol. 3, no. 2, pp. 138-149, 2010.

5. Jonghyun Ryu, Jaehoon Jung, Gunhyuk Park, and Seungmoon Choi, "Psychophysical Model for Vibrotactile Rendering in Mobile Devices", *Presence: Teleoperators and Virtual Environments*, vol. 19, no. 4, pp. 364-387, 2010.

6. Sangki Kim, Gunhyuk Park, Sunghoon Yim, Gabjong Han, Seokhee Jeon, Seungmoon Choi, and Seungjin Choi, "Gesture-Recognizing Hand-Held Interface with Vibrotactile Feedback for 3D Interaction", *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1169-1177, 2009.

International Conferences

1. Gunhyuk Park and Seungmoon Choi, "PhysVib: Physically Plausible Vibrotactile Feedback Library to Collisions on a Mobile Device", Asia Haptics, 2016 (Demonstration).

2. Gunhyuk Park, Yongjae Yoo, Seungmoon Choi, Changdo Song, Minjoo Cho, Giuyeol Kim, Jaehun Kim, Sangmin Lee, Kyogu Lee, "TouchMusic: Music Experience System for the Hearing-Impaired," *In Proceedings of the IEEE World Haptics Conference (WHC)*, 2015 (Demonstration).

3. Gunhyuk Park, Seungmoon Choi, Kyunghun Hwang, Sunwook Kim, Jaecheon Sa, and Moonchae Joung, "The Dynamic Haptics Library," *In Proceedings of the IEEE World Haptics Conference (WHC)*, 2013 (Demonstration).

4. Jaemin Chun, Gunhyuk Park, Seungwhan Oh, Jongman Seo, In Lee, Seungmoon Choi, and Sung H. Han, "Evaluating the Effectiveness of Haptic Feedback on a Steering Wheel for BSW," *In Proceedings of the 4th Inter-*

national Conference on Applied Human Factors and Ergonomics, pp. 2047-2054, 2012.

5. Gunhyuk Park and Seungmoon Choi, “Perceptual Space of Amplitude-Modulated Vibrotactile Stimuli”, *In Proceedings of the IEEE World Haptics Conference (WHC)*, pp. 59-64, 2011 (Finalist for the Best Student Paper Award)

6. Gunhyuk Park, Seungmoon Choi, Kyunghun Hwang, Sunwook Kim, Jaecheon Sa, and Moonchae Joung, “Tactile Effect Design and Evaluation for Virtual Buttons on a Mobile Device Touchscreen,” *in Proceedings of International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pp. 11-20, 2011.

7. Jaemin Chun, Gunhyuk Park, Seunghwan Oh, Jongman Seo, In Lee, Seungmoon Choi, Sung H. Han, Woochul Park, “Development of Human Factors Design Guidelines for Haptic Collision Warning Systems,” *In Proceedings of the IIE Asian Conference*, pp. 249-254, 2011.

Domestic Journals

1. Sangki Kim, Gunhyuk Park, Seokhee Jeon, Sunghoon Yim, Gabjong Han, Seungmoon Choi, and Seungjin Choi, “HMM-based Motion Recognition with 3-D Acceleration Signal,” *Journal of KIISE: Computing Practices and Letters*, vol. 15, no. 3, pp. 216-220, 2009

Domestic Conferences

1. Gunhyuk Park, Hojun Cha, and Seungmoon Choi, “Attachable Haptic Device and its User Studies,” *In Extended Abstracts of HCI Korea*, pp. 120-122, 2016.

2. Gunhyuk Park, Kyunghun Hwang, Sunwook Kim, Jaecheon Sa, Moonchae Joung, Seungmoon Choi, “Button Click Modeling using Vibrotactile Feedback on Touchscreen Mobile Phone,” *In Proceedings of the HCI Korea*, pp. 254-256, 2011.

3. Gunhyuk Park, Seunghwan Oh, Hojin Lee, and Seungmoon Choi, “Color Information Transfer with Vibration Feedback,” *In Proceedings of KIISE Fall Conference*, pp. 58-59, 2010.

4. Gunhyuk Park, Sangki Kim, Sunghoon Yim, Gabjong Han, Seungmoon Choi, Seungjin Choi, Hongjun Eoh, Sunyoung Cho, “Improved Motion-Recognizing Remote Controller for Realistic Contents,” *In Proceedings of the HCI Korea*, pp. 396-401, 2009.

5. Seokhee Jeon, Sangki Kim, Gunhyuk Park, Gabjong Han, Sungkil Lee, Seungmoon Choi, Seungjin Choi, and Hongjun Eoh, “Motion-Recognizing Remote Controller with Tactile Feedback,” *In Proceedings of the HCI Korea*, Vol. 1, pp. 1-6, 2008 (a final candidate for the best paper award).

6. Sangki Kim, Gunhyuk Park, Seokhee Jeon, Sunghoon Yim, Gabjong Han, Seungmoon Choi, and Seungjin Choi, “HMM-based Motion Recognition with 3-D Acceleration Signal”, *In Proceedings of KIISE Fall Conference*, pp. 69-70, 2008 (Winner of the best paper award)).

7. Jaehoon Jung, Inwook Hwang, In Lee, Chaehyun Lee, Gunhyuk Park, Jane Hwang, Seungmoon Choi and Gerard J. Kim, “Remote Control for Motion-Based Interactions,” *In Proceedings of the HCI Korea*, pp. 115-122, 2007.