HapticPilot: Authoring In-situ Hand Posture-Adaptive Vibrotactile Feedback for Virtual Reality

YOUJIN SUNG, KAIST, Republic of Korea
RACHEL KIM, KAIST, Republic of Korea
KUN WOO SONG, KAIST, Republic of Korea
YITIAN SHAO, HITSZ, China
SANG HO YOON, KAIST, Republic of Korea

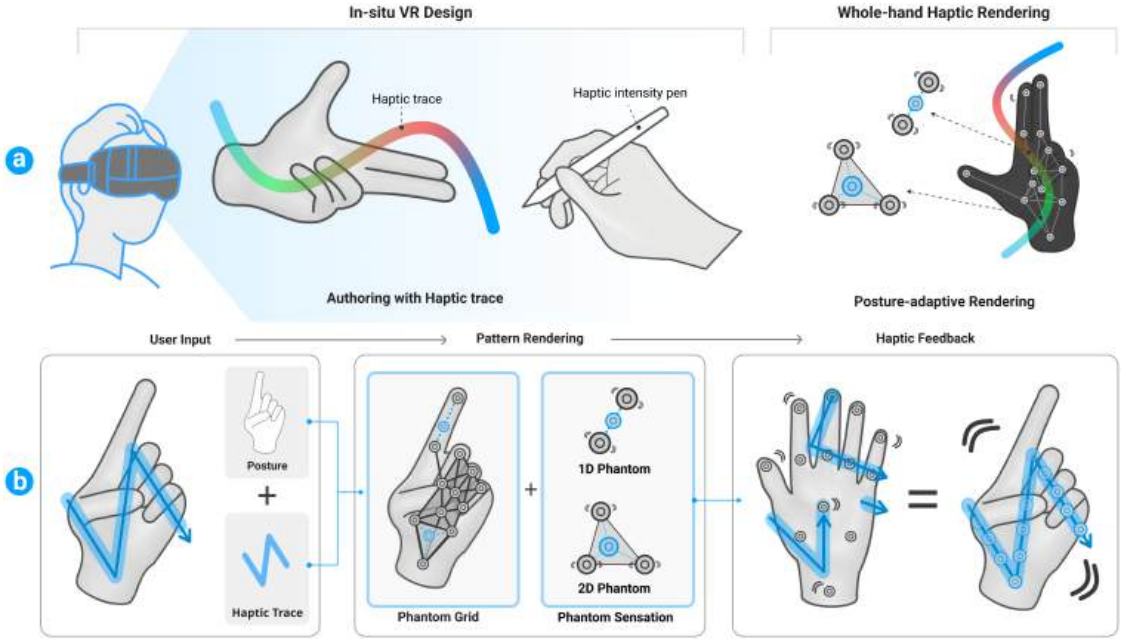

Fig. 1. (a) HapticPilot is an in-situ haptic authoring framework with a posture-adaptive algorithm. (b) With given haptic traces, it maintains the intended haptic experience over different hand postures using phantom grid and integrated 1D & 2D phantom sensation.

Authors' addresses: Youjin Sung, 672@kaist.ac.kr, KAIST, 291, Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea, 34141; Rachel Kim, rachel02@kaist.ac.kr, KAIST, Daejeon, Republic of Korea, 34141; Kun Woo Song, kwsong0725@kaist.ac.kr, KAIST, 291, Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea, 34141; Yitian Shao, shaoyitian@hit.edu.cn, HITSZ, Shenzhen, China; Sang Ho Yoon, sangho@kaist.ac.kr, KAIST, 291, Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea, 34141.

2474-9567/2023/12-ART179 $15.00
https://doi.org/10.1145/3631453

The emergence of vibrotactile feedback in hand wearables enables immersive virtual reality (VR) experience with whole-hand haptic rendering. However, existing haptic rendering neglects inconsistent sensations caused by hand postures. In our study, we observed that changing hand postures alters the distribution of vibrotactile signals which might degrade one's haptic perception. To address the issues, we present HapticPilot which allows an in-situ haptic experience design for hand wearables in VR. We developed an in-situ authoring system supporting instant haptic design. In the authoring tool, we applied our posture-adaptive haptic rendering algorithm with a novel haptic design abstraction called phantom grid. The algorithm adapts phantom grid to the target posture and incorporates 1D & 2D phantom sensation with a unique actuator arrangement to provide a whole-hand experience. With this method, HapticPilot provides a consistent haptic experience across various hand postures is available. Through measuring perceptual haptic performance and collecting qualitative feedback, we validated the usability of the system. In the end, we demonstrated our system with prospective VR scenarios showing how it enables an intuitive, empowering, and responsive haptic authoring framework.

CCS Concepts: • **Human-centered computing → Systems and tools for interaction design**; **User interface toolkits**; **Haptic devices**; **Virtual reality**.

Additional Key Words and Phrases: haptic experience, haptic design authoring tool, posture-adaptive algorithm, phantom grid

ACM Reference Format:
Youjin Sung, Rachel Kim, Kun Woo Song, Yitian Shao, and Sang Ho Yoon. 2023. HapticPilot: Authoring In-situ Hand Posture-Adaptive Vibrotactile Feedback for Virtual Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 179 (December 2023), 28 pages. https://doi.org/10.1145/3631453

1 INTRODUCTION

In virtual reality (VR), incorporating haptic feedback with hand interaction has shown significant promise in the sense of embodiment [6], task performance [54], and immersion [58]. Since the hand is a key interaction medium in VR [56, 59], haptic researchers put a high priority on simulating the sensation of hand touch [40, 50]. Among various approaches, they came up with wearable tactile feedback devices [19, 21]. To avoid interfering with hand movement, compact and lightweight haptic gloves incorporated with vibrotactile actuators were often used [7, 9, 30, 53].

Still, vibration signals produced by the actuators can be influenced by hand movement resulting in inconsistent haptic experience for users. The haptic perception of tactile sensation can easily alter based on the associated hand postures [27]. For example, even a small change in hand configuration can easily affect a human's perceived sensation of vibrotactile feedback [61]. This is because hand postures affect the pathway where a single tactile stimulus (vibration) propagates as well as alter the direction in how a sequence of stimuli progresses. We observed confusion of tactile direction and distance due to changes in hand postures in our exploratory studies. Without considering hand postures, the haptic feedback elicits incorrect sensations, harming both tactile localization performance and VR immersion.

Previous haptic rendering approaches with hand mainly focused on providing the realistic haptic sensation [22, 25, 30, 42, 58] while accommodating various hand postures is often ignored despite of anticipated degradation in VR immersion. These works could only support limited spatiotemporal haptic experiences, like providing discrete sensations on different hand locations. This is a huge drawback in VR since hand interactions have gained increasing attention with high expressiveness and control flexibility [56, 59].

To this end, we propose a hand posture-adaptive haptic rendering algorithm that automatically translates a given haptic design pattern to suit a set of hand postures. The phantom grid can determine a user's hand status by tracking its components and boundaries (*e.g.*, hand boundary, palm boundary), which incorporates the spatial information of the whole hand. We employed 1D and 2D phantom sensation with a unique actuator arrangement to respond to all possible areas covered by the phantom grid. With the proposed phantom grid and phantom sensation approaches, our tool enables haptic rendering for various hand postures. Even with various

hand postures, our posture-adaptive rendering algorithm achieves comparable information transmission (IT), recognition accuracy, and similarity scores in our user studies with 20 participants.

Despite advances in haptic rendering methods, the current design process still falls short of providing effective tactile sensations for changing hand postures. It requires haptic designers (or hapticians) to manually design separate tactile feedback patterns for different postures to maintain similar haptic experiences [3, 46]. In this paper, we introduce HapticPilot, a haptic design tool enabling hand posture-adaptive vibrotactile feedback design and rendering for VR hand interactions. To the best of our knowledge, no studies have suggested a haptic design tool that accommodates various hand postures when prototyping vibrotactile feedback for VR hand interactions.

Our authoring framework supports instant playback and customization in VR to maximize the flexibility and quality of the haptic experience design process. We conducted user studies to validate the performance and system usability of our posture-adaptive haptic design tool, including perceptual tests and in-depth interviews. In the authoring system usability evaluation, we collected qualitative feedback and an overall SUS score of 81. In-depth interviews revealed that design through in-situ sketching and supporting various hand postures was efficient and user-friendly. The overall results showed that our approach is empowering, intuitive, and responsive in prototyping vibrotactile feedback for VR hand interactions.

HapticPilot reduces designers' workload for iterative design processes. The proposed posture adaptive rendering method removes the need to manually edit vibrotactile patterns for each different hand posture. This supports rich and robust haptic design for the whole hand regardless of hand postures while supporting instant design modification. Thus, it would allow HCI practitioners to design a more immersed VR experience.

Our contributions are as follows:

- We identified how hand postures change the vibrotactile feedback patterns via physical measurement. Our psychophysical study indicated that such changes negatively affect the users' haptic experience.
- We proposed a novel haptic design abstraction called phantom grid considering the spatial information of hand anatomy and actuator location to enable posture-adaptive haptic rendering.
- We devised a unique actuator arrangement incorporating line (1D) and polygon (2D) phantom sensation that supports rich vibrotactile patterns covering the whole hand with the proposed phantom grid.
- We developed HapticPilot which is an in-situ VR haptic design tool that supports an intuitive, empowering, and responsive haptic design framework based on our versatile haptic rendering technique.

2 RELATED WORK

Embedded with thousands of mechanoreceptors [15], hands are the primary receiver of haptic feedback among various body parts. To create a robust and rich haptic experience with VR hand interactions, several key aspects must be considered simultaneously. Here, we discuss these three key areas that are essential in enabling a successful haptic experience in VR.

2.1 Haptic Interface for Hand

The sense of touch is essential in acquiring tactile information to carry out a task [49] in everyday experiences, from picking up a cup to typing the keyboard. Considering a wide range of tactile cue perception [50], researchers employ haptic feedback directly on the hand with various haptic interfaces [8, 13, 43]. However, these haptic interfaces may constrain or interfere with hand movement. Thus, researchers started to use vibrotactile actuators as they are lightweight, low-cost, and small.

To transfer vibrotactile feedback to users' hands, low-cost and small form factor commercial haptic gloves have been introduced [3, 48]. However, due to insufficient utilization of tactile illusions and heavy reliance on single-point vibrations, these gloves support tactile feedback to the hand with limited spatial coverage. Recently, researchers proposed glove-based whole-hand vibrotactile feedback [7, 9, 30, 36, 53] to provide direct hand

stimulation and versatile interactive experiences. These works show improved spatiotemporal coverage and perceived performance of tactile sensation on whole hands. Still, previous works assume the static hand posture for rendering tactile sensation, limiting feasibility and scalability.

The haptic perception of tactile sensation altered easily upon changing hand postures [27]. For example, even a small change in hand configuration can disturb a human's perceived sensation of vibrotactile feedback [61]. This is because hand postures affect the pathway where a signal (vibration) from a single tactile stimulus propagates as well as changes the direction in which a sequence of stimuli progresses. Without considering hand postures, the haptic feedback often stimulates incorrect sensations, harming both perceptual recognition performance and VR immersion. In our work, we qualitatively identified how changes in hand postures affect the haptic experience by examining physical and psychophysical performances. Based on the results, we came up with a unique arrangement of actuators on the hand that helps maintain the tactile sensation for different hand postures with the proof-of-concept gloves consisting of 13 LRAs.

2.2 Tactile Rendering for Hand

Among various haptic rendering approaches, we utilized the phantom phenomenon, which refers to the use of distant tactile stimulation to create a spatial tactile illusion between two or more nodes [1]. Since phantom sensation requires a small number of actuators, it causes less disturbance during hand interactions in VR while delivering rich spatiotemporal vibrotactile feedback. Previously, researchers actively employed phantom sensation on hands [9, 33, 37, 39, 41, 42]. More recently, researchers also extended phantom phenomena deployment onto the whole hand utilizing line (1D) and polygon (2D) based phantom sensations [14, 28, 38]. In this work, we further extended previous work by integrating 1D & 2D phantom sensations to render rich vibrotactile feedback covering the whole area of the hand.

The above-mentioned works still assume static hand posture conditions where the designed phantom sensations are fragile upon the user's hand posture changes. To this end, we propose a posture-adaptive haptic rendering algorithm. As a basis of posture-adaptive rendering, we propose a new haptic design abstraction called phantom grid which represents spatial information of hand structure and attached motors. Then, we construct a versatile haptic rendering method that is capable of generating vibrotactile feedback patterns that feel similar across distinctive hand postures.

2.3 Haptic Design in HCI

With the rise of multisensory interactions in HCI, researchers suggest a theoretical model of factors for understanding and evaluating haptic experience [20]. Still, there are many challenges to the haptic design process [47] such as the need for real-time feedback and direct modification on demand [20, 47]. In addition, perception of any given haptic design is different per individual [10] and context dependent [17]. Considering these challenges, a haptic design framework should support instant design iterations [44] and cope with a haptic hardware platform to support on-demand design modifications [16].

To support an interactive haptic design process, recent works showcase a responsive web-based authoring interface [46], trace-based vibrotactile feedback design [3], and sketching on 3D body [4]. These methods support instant haptic feedback and fine-tuning control panels, allowing quick design iterations. However, a 2D desktop GUI-based toolkit makes it hard to achieve in-situ haptic design modifications for VR hand interactions since users need to switch between the desktop and VR interface to design a haptic effect. As such, recent works suggest in-situ prototyping for a VR environment with direct manipulation [5, 12]. Inspired by these works, we further extend an in-situ VR haptic design tool where users directly design vibrotactile feedback patterns on hand solely based on hand interactions.

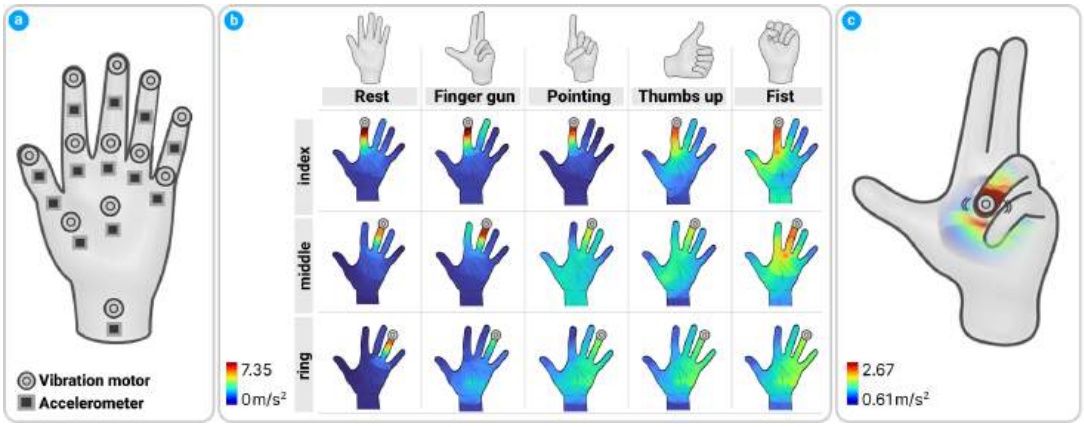

Fig. 2. We measured spatial distribution of whole-hand skin vibration from (a) 12 hand regions using 13 accelerometers on five hand postures. (b) The five hand postures and their respective heatmap of the distribution of motor-elicited skin vibrations when a motor was attached on the index, middle, and ring finger. (c) It shows skin vibration propagation from the finger to the palm when you make finger gun posture.

3 DESIGN OF HAPTICPILOT

In this section, we report the development and design details of **HapticPilot**, a posture-adaptive haptic design toolkit that alleviates the mental and physical workload from hapticians (Figure 1).

3.1 Haptic Performance on Hand Postures

To understand the need for a posture-adaptive toolkit, we carried out two exploratory studies testing how various hand postures affect physical and psychophysical performance for whole-hand vibrotactile feedback. For the physical performance, we collected spatial distribution of whole-hand skin vibrations for a given vibrotactile stimulus during various hand postures. For the psychophysical aspect, we measured IT in bits [52] to confirm the information transmission capacity change upon hand postures. The changes in hand postures, specifically those that involved more skin contact area, resulted in distortions of the original vibrotactile feedback.

For the two exploratory studies (Figure 2a), we directly attached 12 linear resonant actuators (LRA) motors (VG1040003D, Vybronics) operated with motor drivers (DA7280, Dialog Semiconductor). We applied adhesive tape (468MP, 3M) to attach electrical components to the skin throughout the analysis. We attached stimulus tactors on the palmar side of the non-dominant hand (left-hand) which is more sensitive to somatosensory feedback [2, 24]. We picked five commonly used hand postures in VR [11, 31, 32, 51] including *Rest*, *Finger Gun*, *Thumbs Up*, *Fist*, and *Pointing* (Figure 2b).

3.1.1 Exploratory Study 1: Spatial distribution of whole-hand skin vibrations. For physical performance, we examined the propagation of skin vibrations throughout the hand [34].

Setup We recruited 10 participants (4 females, 6 males) with a mean age of 22.9 who were all right-handed. We attached 13 accelerometers (ADXL335, Analog Devices). For each motor, driven with an input voltage of 2.5 Vrms and a frequency of 170Hz, the vibrations (in m/s^2) were measured from 13 accelerometers. Vibration intensity was measured with the root mean square (RMS) value of the data acquired for 1.5 seconds at a sampling rate of 4.4 kHz. We repeated these three trials for all five hand postures to collect a total of 23,400 data points (10 participants × 5 hand postures × 12 motors × 13 sensors × 3 trials).

Result From the RMS value of acceleration, we obtained a skin vibration heatmap for the whole hand, similar to [55]. We observed increased area and amplitude of skin vibrations as the hand posture involved more skin contact. The skin contact points serve as extra pathways for vibration propagation (Figure 2c). We noticed that the posture with a high occurrence of skin contact tends widespread skin vibrations. Here, the skin contact area increases in the order of *Rest*, *Finger Gun*, *Pointing*, *Thumbs Up*, and *Fist* postures. For more details on the result, please refer to **Appendix A**.

3.1.2 Exploratory Study 2: Information Transfer. In study 2, we examined the baseline psychophysical performance of applying actuators on overall hand locations. We measured IT when actuating a single motor on different hand locations under various hand postures. The results show that hand postures reduce users' psychophysical performance on vibrotactile feedback.

Setup We recruited another 10 participants (6 females, 4 males) with a mean age of 23 who were all right-handed. We used the same hardware setup as the previous study, excluding accelerometers. To prevent getting any auditory or visual clues, participants wore a noise-canceling headset that played white noise, and the display monitor separated their left arm. To measure IT, we activated 12 motors one at a time for 3 seconds in random order and repeated this for all five hand postures. After a stimulus was played, we asked participants to choose the location where they felt vibration. Participants responded using a mouse click with a given GUI showing the 12 motor locations. We repeated this procedure 6 times. A total of 46,800 data points were collected (10 participants $\times$ 6 trials $\times$ 12 motors $\times$ 5 hand postures $\times$ 13 sensors). To minimize fatigue, a 5-minute break was given after every 3 trials and extra breaks upon request.

Results *Rest* had the highest IT with 2.92 followed by *Finger Gun*(2.55), *Pointing*(2.44), *Thumbs Up*(2.40), and *Fist*(2.10). Non-*Rest* hand postures exhibit relatively low IT. Among them, *Fist*, which had the most skin contact between the fingers and the palm exhibited the lowest IT (2.10 bits). This signifies that the widespread vibration propagation negatively affects the user's haptic perception. To prevent these problems, hapticians should create haptic designs with the corresponding postures in mind. This would subsequently increase design task loads since the designers need to compute the distortion of motor vibration upon skin contact constantly.

3.2 Pilot Study on Early Prototype

After examining the physical and psychophysical aspects, we conducted a small pilot study with novice hapticians [47] to determine the design consideration for an in-situ haptic design tool. We implemented primary features for conventional in-situ haptic design tools including hand tracking-based haptic design, instant customization in VR, and hand posture-adaptive compensation using skin vibration heatmaps.

Setup & Procedure We recruited 12 novice hapticians (7 females, 5 males) with a mean age of 23.3 who were all right-handed. Participants were equipped with a VR headset (Meta Quest Pro) and 12 LRA motors (Figure 2a). After a five-minute practice session for the user interface and its functions, we asked participants to design vibrotactile feedback to match the given six visual effects (VFX) on five hand postures. We encouraged an iterative design process without a time limit to collect in-depth feedback from participants. After the entire design process was over, we obtained qualitative feedback from participants on the overall design experience as well as the vibrotactile feedback itself. The study took about one hour for each participant.

To support in-situ haptic design, we provided an interface where participants could edit and play the designed vibrotactile feedback instantly inside the VR environment. The participants evaluated two distinct toolkits with different rendering approaches. In the first approach (conventional method), users had to manually select the motors on their virtual left hand to create a vibrotactile sequence over time. They were given a corresponding skin vibration heatmap for haptic design assistance when they hovered around the motor region. In the other approach (automated method), participants designed by coloring their left hand to designate the target stimulation

area over time. Based on the colored region, the system determines which motor to activate in consideration of the skin vibration heatmaps so that the colored region stimulates prominently.

Results Similar to [5], participants gave positive comments on the automated design. However, participants sought more detailed customization features such as controlling vibrotactile intensities and sequences. Regarding the haptic rendering implementation, participants preferred the automated design approach which reduces designers' workloads. *"I felt much-reduced task load since I did not have to care about which motor to activate (P10)"*. In the same context, participants barely utilized the provided heatmaps since it increased task complexity. We also observed a drawback of a hand-tracking-based tool due to frequent tracking loss during hand overlapping. Besides, the design of some participants exceeded the drawable area of our vibrotactile glove, causing a perceptual discrepancy. By assessing the proof-of-concept toolkit, we enhanced the design of our toolkit.

3.3 Exploratory and Pilot Studies Implications

Based on the cornerstone from the above studies, we outline design considerations for designing HapticPilot which is an in-situ haptic design tool for VR hand interactions.

Detailed customization support for in-situ VR authoring With the advancement of the VR authoring environment, recent works have shown improved performance and benefits using in-situ VR haptic design process [5] over conventional desktop GUI approach [3]. We also received positive feedback from participants. *"It is intuitive to design haptic feedback directly on the hand while watching VFXs in VR (P8)"*. Furthermore, we discovered users' demand for more detailed spatiotemporal customization of the vibrotactile feedback design. To be more specific, participants preferred controlling the intensity of the vibrotactile feedback and previewing the designed pattern with animated traces.

Balancing between design freedom and system robustness Designing a haptic experience is a complicated task where spatial and temporal elements should be considered simultaneously. With numerous vibrotactile stimuli, the design task gets complicated and restricts design creativity. To support similar design freedom as the previous work [18], our system allowed users to color the virtual hand using natural hands. Unlike previous works, it requires no controllers where all interactions are done using hand tracking. Participants showed positive feedback on utilizing the concept of sketching. *"It was convenient to design haptic feedback through coloring with natural hands compared to selecting and adjusting motors" (P4)*. Several participants, however, showed concerns about the system's robustness related to hand tracking. *"VR hand tracking seems fragile when two hands are in contact or crossing each other" (P2)*. Thus, the prospective system and interface should support a hand-tracking-based tracing interaction with high robustness.

Automated and adaptive haptic rendering for hand postures During the pilot study, participants reported hardship in manually designing vibrotactile feedback with various hand postures in mind. For different hand postures, designers must maintain spatial relationships while preventing unexpected vibration propagation due to skin contact. Although the system provided a supplementary skin vibration heatmap for design guidelines, participants still reported a high workload in using it for the haptic design. *"I had no idea how to utilize the given skin vibration heatmap for the haptic design" (P12)*. Also, participants exhibited fatigue in the overall design process since they had to iterate the same design process for each hand posture. To resolve these issues, we embed a posture-adaptive haptic design algorithm in our design toolkit. This allows an automated design process across various hand postures while reducing hapticians' workloads.

3.4 Overview of HapticPilot Interface

Considering implications from the above studies, we developed an in-situ VR authoring framework that supports 1) detailed customization, 2) high design freedom with a robust system, and 3) automated posture-adaptive haptic rendering.

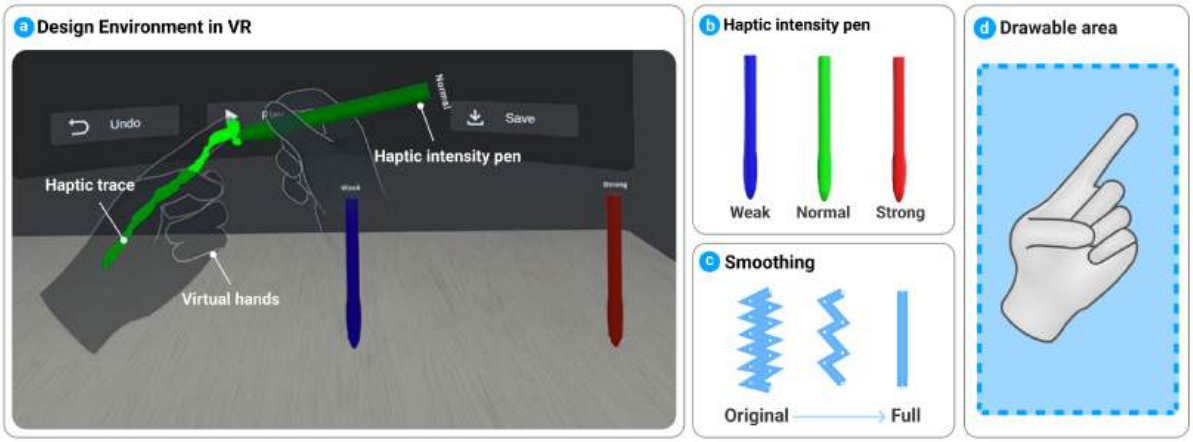

Fig. 3. HapticPilot interface overview. (a) A user sketches a haptic trace with a (b) haptic intensity pen on the virtual hand. Upon user sketches, the system collects haptic points, which can be used for (c) smoothing haptic traces if needed. (d) A drawable area refers to the region around the hand to draw haptic traces.

We developed HapticPilot using the Unity3D Engine (2021.3.6f1) and Meta Quest Pro HMD. To facilitate a controller-less design environment, we adopted Oculus Interaction SDK [35] to support basic hand tracking and interactions such as raycasting & pinch-based pointer pose. To avoid hand tracking loss due to occlusion, we came up with virtual haptic intensity pens (Figure 3b). Employing a haptic intensity pen in VR as a tool to draw haptic traces on the hand improved the control robustness since this method naturally prevents hands from overlapping. We also provided multi-color pens to denote different vibrotactile intensities (Blue-Weak, Green-Normal, and Red-Strong).

When users initiate sketching with the haptic intensity pen on the virtual hand, our system collects sketching points at 100 Hz to form haptic traces. As shown in Figure 3c, the system smooths out traces after each stroke, which is done by taking sketching points at every constant interval determined by the user. Our tool provides features for detailed customization support such as 'Undo haptic traces', 'Instant haptic playback', 'Edit pattern', and 'Save'. Furthermore, we embedded the posture-adaptive algorithm (**Section 3.6**) that automatically modifies haptic traces upon different hand postures, so users do not need to carry out the same haptic design procedures.

HapticPilot Walk-through Here, we explain a step-by-step procedure for HapticPilot. First, the designer chooses a posture to start the haptic design process. Then, the virtual hand gets fixed to the selected hand posture for users to sketch. We supported five hand postures, including *Rest*, *Finger Gun*, *Thumbs Up*, *Fist*, and *Pointing*. The designer can sketch multiple independent traces that can be spatially overlapped, but to keep haptic rendering precision, they cannot be actuated simultaneously. In the early prototype pilot study (**Section 3.2**), we noticed that participants often drew continuous traces extending outside the hand region. To provide more design freedom for users, we set the boundary of the canvas to span beyond the hand (Figure 3d). The system supports an instant playback feature where haptic traces are animated on an empty virtual hand (Figure 3a~c). The animated haptic traces are synced with actual hardware, so hapticians can easily experience designed vibrotactile feedback in-situ. Lastly, the haptic design can be saved by selecting the save button, which automatically generates vibrotactile patterns for different hand postures by applying our posture-adaptive algorithm to the created design (**Section 3.6**). By selecting different hand postures from the UI panel, the designers experience

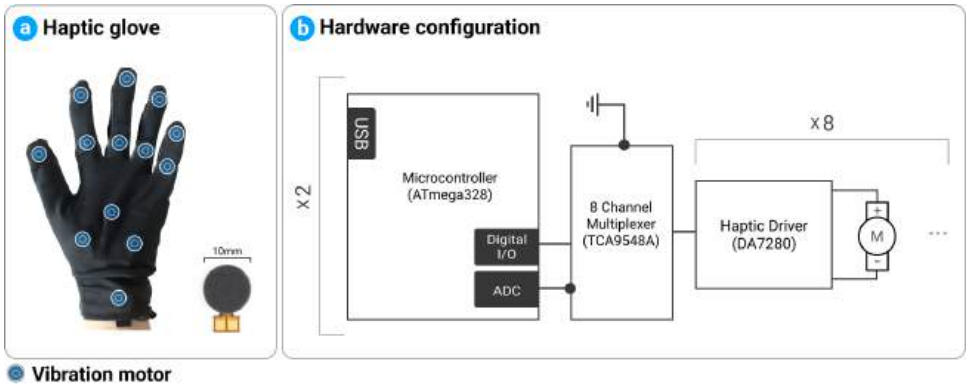

Fig. 4. Our (a) proof-of-concept haptic glove embedded with 13 LRAs and (b) hardware configuration.

the automatically modified haptic design. Furthermore, we allow hapticians to edit the haptic design for each posture if needed.

3.5 Hardware

To design our own haptic glove that creates rich sensations but enhances users' movement, we developed a haptic glove constituting 13 motors (Figure 4a). To determine the number of motors, we looked into the required distance between motors. The motors should be close enough to apply phantom sensation while securing some distance to avoid any disturbance and discomfort during hand movements. Considering the above factors and referring to existing literature [29, 39], we placed the motors at least 15 mm away from each other.

We used sports gloves (polyester 85%, polyurethane 15%) which are spandex-based and tightly fit the hand to maintain the shape of the hand for robust hand tracking using VR head-mounted display (HMD). It is available in two different sizes (medium & large). 13 LRAs (VG1040003D, Vybronics) are attached inside of the glove using double-sided adhesive (3M, 468MP)). We operated two Sparkfun Redboards (Atmega328) with 8-to-1 multiplexers (TCA9548A, Texas Instrument) to control a total of 13 haptic drivers (DA7280, Dialogue) (Figure 4b). We kept enough length (≥30 cm) for interconnecting wires to maintain robust connections during hand movements. The latency from command to actuation trigger was less than 4 ms which is not noticeable, and the maximum frame rate was 250 Hz. We operated the development and user studies using a desktop PC with a 3.2 GHz Intel Core i9 processor.

3.6 Posture-Adaptive Algorithm

In this section, we introduce our posture-adaptive algorithm that enables various haptic feedback for different hand postures. We first created a haptic design abstraction called a phantom grid which contains spatial information about the hand. We combined 1D and 2D phantom sensation techniques with a unique actuator arrangement to enable a rich haptic experience covering the whole hand. We developed haptic rendering methods in accordance with the phantom grid of the target posture to maintain similar vibrotactile sensations under different hand postures.

3.6.1 Phantom Grid Construction. The phantom grid is a novel haptic design abstraction consisting of nodes and lines that incorporate the spatial information of the whole hand. The main role of the phantom grid is to create a haptic rendering map to employ 1D & 2D phantom sensations. Figure 5 showcases the major components of

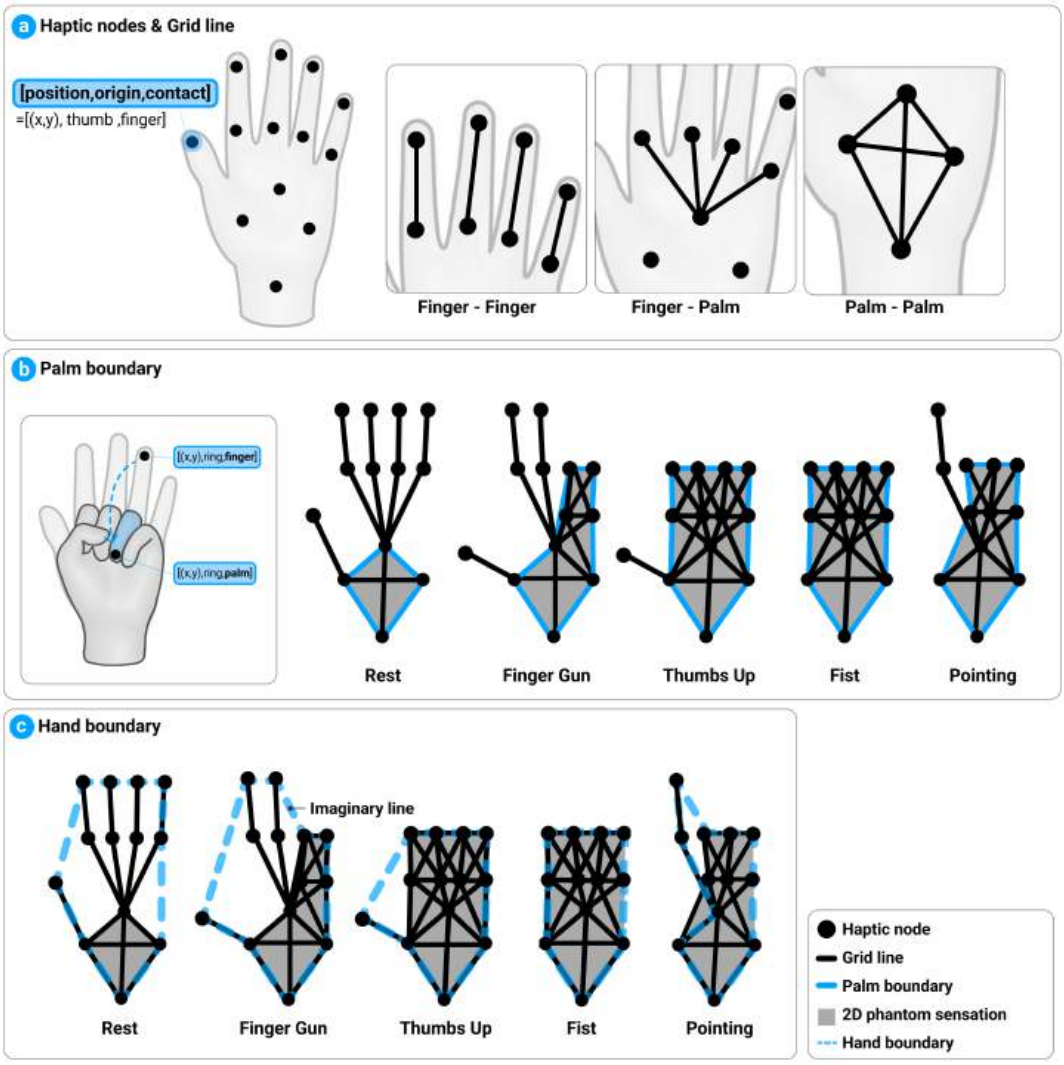

Fig. 5. Overview of phantom grid. (a) A haptic node contains (x,y) *position*, *origin*, and *contact*. Connecting adjacent haptic nodes on the finger and the palm form grid lines. (b) We form palm boundary utilizing the haptic nodes on the palm which expands if the finger is folded for different hand postures. (c) The hand boundary is set to encompass the outermost grid lines, and extra imaginary lines are used to complete the closed hull.

phantom grid construction. It starts by identifying the details of haptic nodes representing the location of 13 LRAs on the hand.

Haptic Node A haptic node (Figure 5a) is a reference point for the phantom grid construction. Each LRA attached to the haptic glove becomes a haptic node consisting of three variables: *position*, *origin*, and *contact*. The *position* refers to the x and y coordinates of the LRA motor's location which varies among hand postures. The *origin* represents the original attached locations of the LRA motors (*e.g.*, thumb, index, middle, ring, little, palm),

and *contact* tells which body area the haptic node contacts with (*e.g.*, finger, palm). Here, *contact* variable may change depending on the finger flexion conditions in different hand postures. For example, on the *Fist* gesture, the haptic node located on the ring fingertip moves closer to the palm center node than the node located on the proximal phalanx. Then, this changes the *contact* variable from 'finger' to 'palm' as shown in Figure 5b.

Grid Line Grid lines (Figure 5a) are edges connecting two haptic nodes that create a phantom sensation. Two adjacent nodes create a phantom sensation when their *origin* or the *contact* is the same. A grid line is established between any two adjacent nodes that are close enough to allow skin vibrations to propagate from one to the other. Within a finger, a 1D phantom sensation is generated using 2 motors which are enough to cover the entire finger while not obstructing hand movement. We attached motors on both ends (distal phalange and proximal phalange) of the fingers except the thumb. Putting 4 motors on the palm area creates a 1D phantom sensation connecting the motors attached to the finger and palm if the motors are close enough.

Palm Boundary A palm boundary (Figure 5b) designates the region for 2D phantom sensation. By default, four haptic nodes at the palm form the palm boundary. This boundary expands if a finger is folded. Upon finger flexion, fingertip nodes come into contact with the palm, which causes motor-elicited skin vibrations to spread throughout the palm. All these nodes are then considered to be on the palm and utilized for producing a 2D phantom sensation. The palm boundaries for all hand postures are illustrated in Figure 5b. Each boundary encompasses finger haptic nodes that make contact with the palm.

Hand Boundary We defined a hand boundary (Figure 5c) to render the haptic sensation outside the palm boundary. Like the palm boundary, we obtained the concave hull of all the nodes on the hand, which gives different boundaries per hand posture. In contrast to the palm boundary, a hand boundary cannot be made using only the grid lines, so we created imaginary lines (Figure 5c) to complete the concave hull. When a point is mapped to an imaginary line, the nearest motor is activated for rendering. For more details on the phantom grid construction algorithm, please refer to **Appendix B**.

3.6.2 Integrated 1D & 2D Phantom Sensation for Whole Hand. We integrated phantom tactile sensation techniques [14, 39] to enable continuous vibrotactile feedback beyond simple point and line covering whole hand. Here, we employed both 1D & 2D phantom sensation techniques to respond to all possible areas covered by the phantom grid. With a unique actuator arrangement,a phantom sensation for a target location is created by stimulating up to 3 motors (2 motors for 1D & 3 motors for 2D cases). We chose the energy model shown in Eq. 1 [39] which demonstrated the best phantom localization accuracy compared to other models [14, 45].

$$A_{motor} = \sqrt{\frac{1/d_{motor}}{\sum_i 1/d_i}} A_{target} \text{ (A: amplitude, d: distance from a motor to the haptic point)} \quad (1)$$

3.6.3 Posture-Adaptive Haptic Rendering. With the proposed phantom grid with phantom sensation approaches, our tool enables haptic rendering in various hand postures. The users design vibrotactile patterns in the form of 2D traces called haptic traces. These traces consist of points collected at 100 Hz, which we call haptic points. We execute various haptic rendering methods according to the geometric relationship between the haptic point and the phantom grid. The geometric relationship includes haptic node collision, grid line intersection, within-palm localization, and outside-palm localization.

Haptic Node Collision This is the simplest form of haptic rendering. We loop through all haptic nodes and compute the distance between each haptic node and the haptic point. If the point lies within the distance threshold (physical motor size), the system activates the corresponding motor with the target amplitude (Figure 6a).

Grid Line Intersection If the haptic point is located on the grid line, we employ 1D phantom sensation. Since a grid line is created by connecting two haptic nodes, we drive these motors with Eq. 1. The amplitudes of the motors are adjusted based on the position of the haptic point within the grid line (Figure 6b).

Raycasting Algorithm If a haptic point belongs to none of the above, the haptic point would be located either inside or outside of the palm area polygon created by the palm boundary (Figure 6c). We use the raycasting algorithm (Figure 6c) to distinguish between these two. Suppose the number of intersections between the palm boundary and a ray from the haptic point towards the palm center is odd. In that case, the haptic point locates within the palm boundary (Figure 6d) and vice versa (Figure 6e).

Within-Palm Localization If the haptic point locates within the palm, we employ 2D phantom sensation. To determine three haptic nodes to be activated for Eq. 1, we find an enclosing triangle. The algorithm searches for a triangle created by three grid lines that contain the haptic point. As shown in Figure 6d, a single haptic point may have multiple enclosing triangles. Here, we proceed with the first triangle found since the perceived sensation would be the same regardless of the selected triangle with the independent modulation of tactors [14, 39].

Outside-Palm Localization Unlike other conditions, outside-palm localization requires an extra translation phase, the nearest point mapping for the haptic rendering. When a haptic point lies outside the palm (Figure 6e), it is projected to the closer intersection between the hand boundary and the ray used for the raycasting algorithm or its opposite ray (Figure 5c). The hand boundary is formed by as many grid lines as possible, allowing the 1D phantom sensation to create a continuous feeling. If the projected points do not lie on grid lines, they are projected to imaginary lines which actuates the closer haptic node creating the imaginary line (Figure 5d). Moreover, we changed the driving intensity by one-tenth of the original for the outside-palm rendering approach to differentiate from the vibrotactile feedback within actual hand [9].

By iterating through the above geometric relationship conditions, HapticPilot enables posture-adaptive haptic rendering. If the condition is met, the described method renders the haptic point. The algorithm checks the condition for the next geometric relationship if the condition is not met. Here, the proposed algorithm allows adaptive haptic design where the system automatically changes the spatiotemporal vibrotactile pattern and

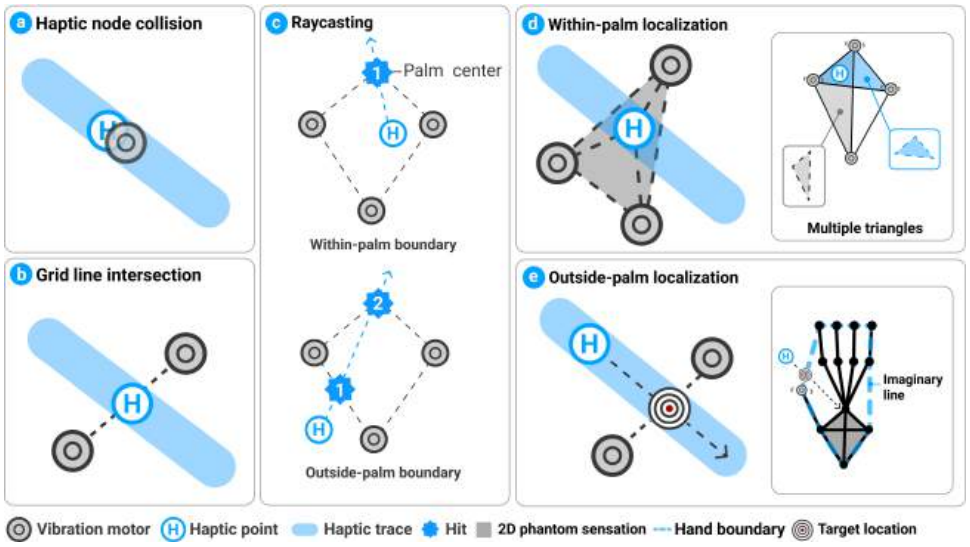

Fig. 6. Our posture-adaptive haptic rendering methods. (a) Haptic node collision activates the motor with which the haptic point overlaps. (b) Grid line intersection uses 1D phantom sensation between two haptic nodes of the grid line. (c) Raycasting algorithm determines whether a haptic point is within or outside the palm. (d) Within-palm localization uses 2D phantom sensation using three haptic nodes forming the enclosing triangle. With independent modulation of tactors, we proceed with the first triangle found. (e) Outside-palm localization projects the haptic point to the nearest hand boundary.

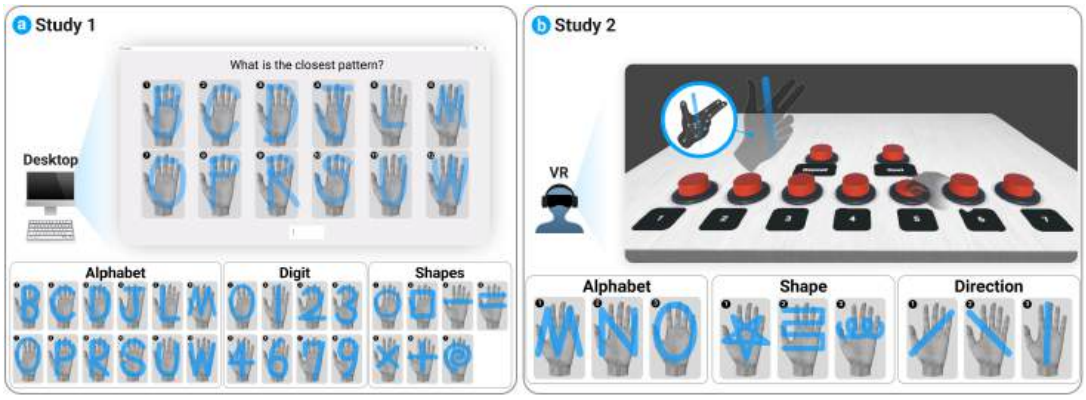

Fig. 7. The overview of study setup and pattern sets. (a) Study 1 is about 1D & 2D phantom sensation recognition for whole-hand and (b) Study 2 is about similarity for posture-adaptive haptic rendering evaluations.

intensities with a given haptic design. Our proposed method aims at preserving perceived tactile sensations for the same haptic traces under different hand postures. For the pseudocode used for posture-adaptive haptic rendering, please see **Appendix B**.

4 HAPTIC EXPERIENCE EVALUATION

As our phantom grid, whole-hand phantom sensation, and posture-adaptive algorithms are novel attempts at rendering haptic feedback, we confirm the base performance of the integrated 1D & 2D phantom sensation with the suggested actuator arrangement and validate the effectiveness of the proposed posture-adaptive haptic rendering. In Study 1, we examine user performance of various whole-hand vibrotactile patterns using an integrated phantom sensation with a suggested actuator arrangement. By obtaining the IT rate and identification accuracy of patterns, we confirm the feasibility of our approach for expressing complex vibrotactile patterns for the whole hand beyond simple point or line-based sensations. In Study 2, we measure the similarity of the same haptic design on different hand postures using the proposed haptic rendering method. By comparing similarities, we validate the robustness of our overall posture-adaptive algorithm across various hand postures.

4.1 Study Design

We recruited 20 participants (8 females, 12 males) with a mean age of 23.4. Participants were all right-handed and their hand size ranged from 15~20 cm. We carried out both studies together which took about two hours. The initial 10 minutes were used for instruction and setting up devices for participants. We provided a 10-minute break between the two studies. We provided an extra break if participants requested which rarely happened.

Apparatus We prepared two sizes of haptic gloves (M&L) since glove-wearing condition affects the performance of vibration transmission on the hand. Before the study, we measured participants' hand sizes and provided either medium ($\leq$ 15 *cm* hand length) or large ($>$ 15 *cm* hand length) size gloves. This guarantees to transmit equivalent vibrotactile sensation to participants with different hand sizes. We applied vibrotactile patterns to a non-dominant hand. Also, participants wore headphones playing white noise during the experiment to prevent any hints from auditory cues like motor sounds. In Study 1, we provided a multiple-choice style desktop GUI with a keypad input to choose the stimulated vibrotactile pattern (Figure 7a). For Study 2, we developed an immersive study environment where participants compare the similarity of given vibrotactile feedback for various hand postures in VR (Figure 7b).

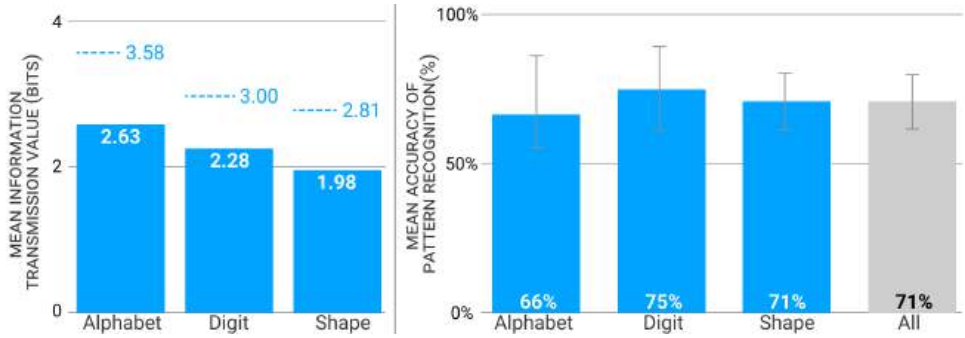

Fig. 8. Study 1 results. (Left) Average IT values for vibrotactile pattern categories (blue bars) with maximum achievable IT (dotted lines). (Right) Pattern recognition accuracy for each category (blue bars) and all categories (gray bar).

4.2 Study 1: Integrated 1D & 2D Phantom Sensation Recognition for Whole Hand

In this study, we validate the performance of integrated 1D & 2D phantom sensation for whole-hand vibrotactile sensation (**Section 3.6.2**). We devised a set of vibrotactile patterns commonly used in VR applications that exhibits more complexity and covers a wide area of the hand compared to previous work [28]. Here, we focused on testing practical vibrotactile patterns, beyond simple point or line-based patterns, that could be employed in VR applications since researchers have confirmed the basic performance of whole-hand phantom sensation [28].

We carefully chose a set of vibrotactile patterns from three categories including alphabets (12 letters), digits (8 numbers), and shapes (lines, circles, rectangle, and curved lines) similar to [23, 57]. To fix the tested patterns, we conducted a pilot test with five participants to exclude those unrelated to the purpose of the study. We excluded alphabets with multiple strokes ('A, E, F, H, I, K, Y') and similar strokes ('C, G', 'J, T, X', 'M, N', 'O, Q', 'S, Z', and 'U, V'). Furthermore, we excluded '5 & 8' which confuse users with different writing styles. We came up with a total of 27 vibrotactile patterns (Figure 7a) with actuation duration up to 1.5 seconds.

Procedure To focus on measuring the performance regarding the vibrotactile sensation only, we asked users to maintain a *Rest* posture. After each stimulus, participants answered patterns from the same category of vibrotactile patterns (alphabets, digits, or shapes). We offered a training session where participants experienced all tested vibrotactile patterns once. A total of 2700 responses were collected (27 patterns × 5 trials × 20 participants).

Results As shown in Figure 8, the IT for alphabet, digit, and shape categories came out to be 2.63, 2.28, and 1.98 bits accordingly, and the overall accuracy was 71% (alphabet: 66%, digit: 75%, shape: 71%). Our IT (1.98 ~ 2.63 bits) and accuracy (71%) came out higher than previous works' [28, 38] 2D stationary phantom IT (2.09 bits) and whole hand location accuracy (71%) even with more complex vibrotactile patterns.

4.3 Study 2: Similarity for Posture-Adaptive Haptic Rendering

In this study, we verify the effectiveness of our newly defined phantom grid (**Section 3.6.1**) and posture-adaptive haptic rendering (**Section 3.6.3**). To do so, we examine similarity which represents how similar the rendered haptic sensation is compared to the ground truth vibrotactile pattern shown as visual traces. We defined similarity as preserving sensed direction and distinguishing within/outside-palm sensation for given vibrotactile feedback. We examined similarity on the four hand postures excluding *Rest* to assess the performance of our posture-adaptive haptic rendering approach. We applied a set of vibrotactile patterns with visual traces in VR in various hand postures (Figure 7).

For Study 2, we chose vibrotactile patterns that cover the entire phantom grid while effectively testing the similarity elements, including the direction and localization (*e.g.*, within or outside-palm) of stimulus. In addition,

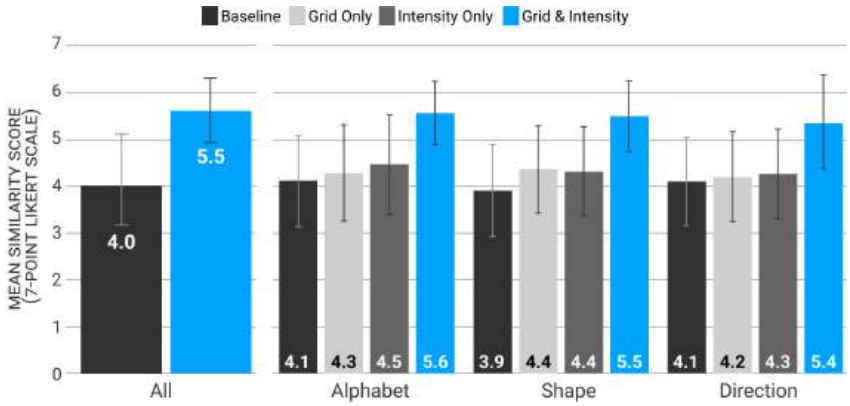

Fig. 9. Study 2 results. (Left) A 7-point Likert scale of similarity for the baseline & full algorithm condition and (Right) all four conditions with three haptic pattern categories. The bars with colors represent different rendering conditions: baseline, grid only, intensity only, grid & intensity (left to right).

we added patterns commonly used in VR applications. Figure 7b illustrates the overall vibrotactile patterns including alphabets ('M', 'N', and 'O'), shapes(star, serpentine line, and helix), and directions (moving towards northeast, northwest, and north).

To explore the validity and efficacy of the proposed algorithm in more detail, we carried out the test under four different conditions as follows:

- **Baseline**: No posture-adaptive haptic rendering was applied.
- **Grid only**: Only the phantom grid was applied to adjust the direction of vibrotactile feedback for hand postures.
- **Intensity only**: Only outside palm localization was applied to differentiate from the vibrotactile feedback within the actual hand.
- **Grid & intensity**: Both phantom grid and outside palm localization was applied.

Procedure In Study 2, participants wore the HMD device (Meta Quest Pro) and proof-of-concept haptic gloves (Figure 7b). The participants were instructed to rate the similarity between the sensed stimulus and visualized traces in VR on a scale of 1 to 7 (Figure 7b) similar to [39]. We asked participants *"How similar was the posture-adaptive tactile experience to the rendered pattern shown in VR?"*. All patterns and postures were tested in random order. A total of 5760 responses were collected (9 patterns × 4 postures × 4 conditions × 2 trials × 20 participants).

Results Figure 9 illustrates Study 2 results. Here, the 'Grid & Intensity' condition with the full algorithm applied showed the highest similarity score of 5.5, significantly larger than the 'Baseline' condition score of 4.0. We observed that applying either a phantom grid or localized intensity modulation did not enhance the similarity score significantly. This implies that phantom grid and localized intensity modulation should be employed together to enhance the similarity. Among various vibrotactile pattern categories, there was a significant increase in similarity score (from 3.9 to 5.5) for the 'shape' category, which consisted of frequent changes in stimulus directions. This indicates that the proposed algorithm is effective in dealing with changes in motor location for different hand postures.

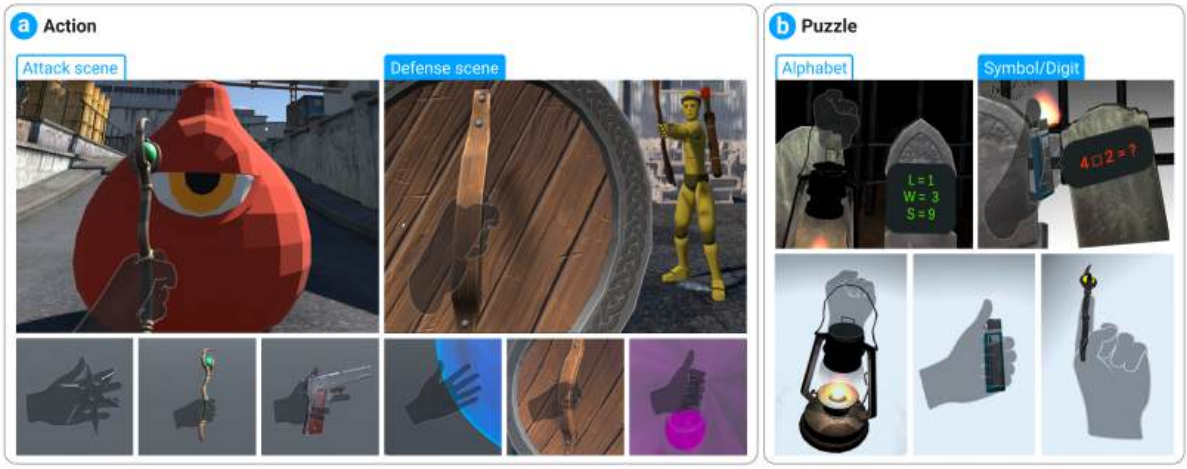

Fig. 10. VR task scenes for system usability evaluation. (a) In *Action* scenes, participants attack or block enemies using different tools, feeling their designs while doing so. (b) In *Puzzle* scenes, participants had to decode a puzzle through their designs using the tools as a light source.

5 AUTHORING SYSTEM USABILITY EVALUATION

In this evaluation, our goal is to understand how hapticians utilize HapticPilot to create meaningful haptic experiences with various hand postures. Participants carried out haptic experience design tasks with and without HapticPilot's full functionality. Then, we compared user experience and preference between these two conditions.

5.1 Study Design

We recruited 20 participants (10 females, 10 males) for the study with a mean age of 25.9 who were all right-handed. 17 participants were researchers/designers experienced in either haptics or human-computer interaction-related fields. We referred to and modified the study design from previous works [5, 39, 56]. Similar to our previous studies, we supported two glove sizes (medium & large). The overall evaluation took up to 90 minutes and we offered training sessions as well as breaks between each task.

Task Scenes We designed scenes for user study tasks under two themes: *Action* and *Puzzle*. We made scenes similar to actual gaming scenes so that hapticians could evaluate the system in real-world scenarios. Figure 10 shows the scenes implemented for the two themes.

For the *Action* theme (Figure 10a), participants designed whole-hand vibrotactile feedback for attack and defense-related VFXs. These two scenes had high design freedom where no specific shapes were enforced while designing vibrotactile patterns. Participants considered *Rest*/*Fist*/*Finger Gun* and *Rest*/*Fist*/*Thumbs Up* postures for attack and defense effects accordingly. We carried out attack and defense scenes in 2 separate sessions.

For the *Puzzle* theme (Figure 10b), participants designed whole-hand vibrotactile feedback to render complicated information like alphabets, mathematical symbols, and digits. We carried out 2 separate design sessions where participants had to create the haptic design for either 5 alphabets ('L', 'W', 'P', 'S', and 'O') or 5 symbols/digits ('×', '+', '2', '3', and '6'). These patterns had to be designed accurately on *Fist*/*Thumbs Up*/*Pointing* postures to solve the puzzles. We carried out alphabet and symbol/digit scenes in 2 separate sessions.

Procedure To explore the usability and confirm the haptic sensation quality using our proposed system, we used the system usability scale (SUS) and an in-depth questionnaire. We created a pseudo-HapticPilot, a version

of HapticPilot without hand posture visualization and posture-adaptive haptic rendering, as a baseline that represents conventional haptic design toolkits [3, 14]. By comparing these two versions of HapticPilot, we expect to understand the effect of adopting hand posture-related features in the haptic design process.

Prior to the study, participants went through a 5-minute training session, followed by instructions on the overall experiment procedure. We expected that the training session would help prevent prior VR experience from affecting the SUS score. Participants practiced hand postures and conducted the Oculus built-in tutorials [56] to get used to the VR environment.

We split 20 participants into 2 groups to counterbalance the order of test conditions. Each group carried out 4 design sessions in total as follows:

- Group1: Pseudo-HapticPilot (*Action* & *Puzzle*) → HapticPilot (*Action* & *Puzzle*)
- Group2: HapticPilot (*Action* & *Puzzle*) → Pseudo-HapticPilot (*Action* & *Puzzle*)

Within each *Action* or *Puzzle* design scene, we randomized and counterbalanced the order of tasks (*e.g.*, *Action*: attack and defense effect & *Puzzle*: alphabet and symbol/digit). Before starting each design scene, participants experienced the scene they would design with haptic design from previous participants (the first participant was exempted from this question).

After each design scene, participants answered the SUS questionnaire and provided qualitative feedback on their design through an in-depth questionnaire. After completing all design sessions, we interviewed the participants for their overall experiences. We only used responses from the last two sessions for SUS analysis to remove potential bias from adaptation and order effect.

Users also completed a 7-point Likert scale questionnaires [56, 60] regarding the quality of haptic sensation and user interaction using the proposed system after each session (Figure 11). We got responses only for the HapticPilot condition since we were interested in user experience with HapticPilot's full functionality. We also asked participants' opinions on a haptic design created by others after the demo for each design scene. They evaluated if *"The designed haptic patterns matched their corresponding hand gestures well"* (Q2 in Figure 11). This enables us to compare how participants comprehend haptic design by others, not just their own.

5.2 Results

5.2.1 Effects of Posture-Adaptive Features. The overall SUS score was 71 (SD=17.3) for the pseudo-HapticPilot condition and 81 (SD=14.2) for the HapticPilot condition. The SUS scores for both conditions were greater than 70, indicating satisfactory usability for the in-situ VR design process with sketching. For the HapticPilot condition, we observed improvement in SUS score from 71 to 81, and more than half of the participants gave a score greater than 80. This clearly indicates that the system usability improved with the proposed posture-adaptive features provided by the HapticPilot.

When using the pseudo-HapticPilot condition without hand posture-related features, participants reported difficulties in designing vibrotactile patterns on a fixed hand posture to accommodate different postures. *"I felt different sensations depending on the hand movements (P7)"*. As a consequence, participants had to make modifications repeatedly from their original intended design. *"When the vibration is applied to fingers, I have to modify it constantly. So I would rather draw it on the palm (P2)"*. With the HapticPilot condition, on the other hand, participants were able to design consistent patterns in various postures. *"I could recognize original patterns on different hand postures (P20)"*. From a design perspective, the HapticPilot framework simplified the complex and repetitive design process to accommodate different hand postures. *"It was convenient to design on different virtual hand postures directly since I did not have to think about changes of vibrotactile sensation due to posture changes. (P2)"*. This also relates to the satisfaction of the output design. *"I edited less because I was satisfied with the automatically generated patterns (P2)"*.

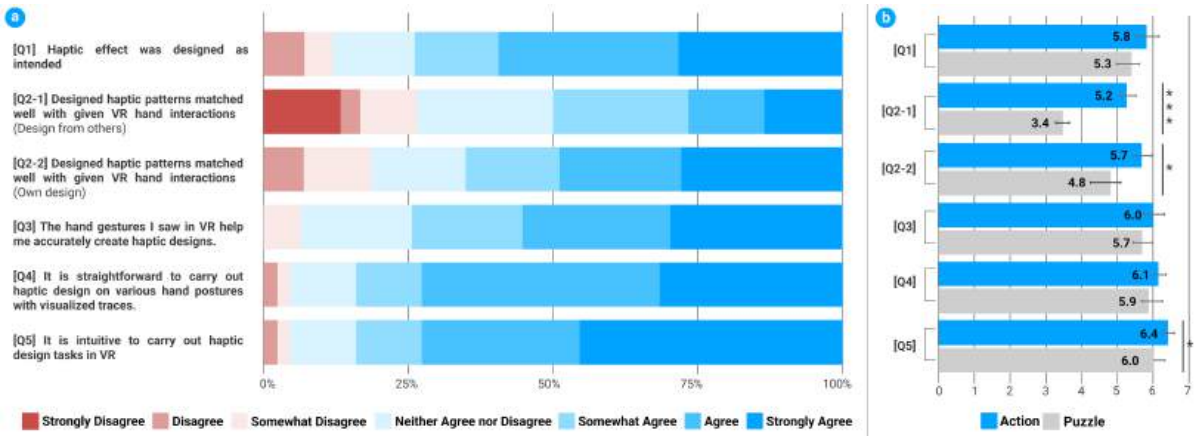

Fig. 11. Authoring system usability evaluation result. (a) The occurrence of responses to questionnaires. (b) Responses to *Action* (blue) and *Puzzle* (gray) scenes.

5.2.2 HapticPilot Design Experience. Figure 11a shows the overall HapticPilot-related Likert-type question ratings. Users generally agreed with the positive effect of employing hand posture-adaptive features for the haptic design process. Users responded that they could design as intended using the system (Q1: AVG=5.5, SD=2.0). Participants were satisfied with the haptic design created with the HapticPilot for different hand postures (Q2-2: AVG=5.3, SD=2.0). Here, we noticed that participants generally favored their own design (Q2-1: AVG=4.3, SD=1.6 vs. Q2-2: AVG=5.3, SD=2.0) due to individual differences in drawing/writing shapes/digits. *"I was confused with others' haptic design since the order of strokes and starting location was different from how I would draw/write it (P15)"*. Moreover, participants responded that having the virtual hand alter the posture throughout the design stage helped them to create accurate designs (Q3: AVG=5.8, SD=1.0). *"It is comfortable to design with hand posture*

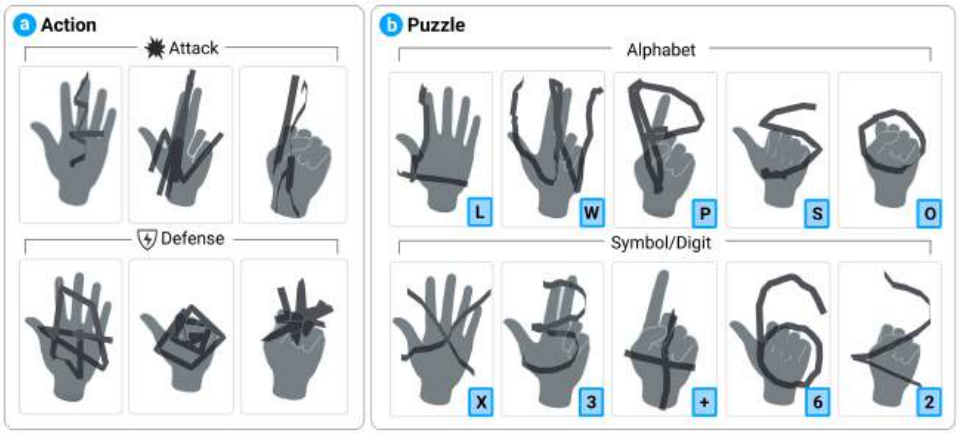

Fig. 12. Haptic design from participants using HapticPilot for (a) *Action* and (b) *Puzzle* scenes. Various design strategies and preferences were observed where some preferred utilizing within-palm regions, and others used wider spaces to focus on rendering the whole shape.

support which makes my design task simpler (P2)". In terms of in-situ design with visualized traces, users reported that it was straightforward to utilize the traces (Q4: AVG=6.0, SD=1.0) and that the in-situ VR design process was intuitive (Q5: AVG=6.2, SD=1.2). *"It is very useful to design while seeing VFX. The VFX helped me confirm that my design worked as intended (P8)".*

5.2.3 Design Freedom. We also compared the Likert rating between *Action* & *Puzzle* task scenes. Figure 11b shows that the proposed hand posture-adaptive features worked better with the task involving more design freedom (*Action* scene). When experiencing other people's designs (Q2-1), users were more satisfied with creative patterns (*Action* scene: AVG=5.2, SD=1.0) over fixed shape patterns (*Puzzle* scene: AVG=3.4, SD=1.0). This aligned with our previous finding that each individual prefers their own writing/drawing style for vibrotactile patterns. We also noticed that users found HapticPilot more intuitive in the *Action* scenes where the design objectives were more abstract. This tendency was statistically observed ($p < 0.05$) from Q5 (*Action*: AVG=6.4, SD=1.0 vs. *Puzzle*: AVG=6.0, SD=1.0).

5.2.4 Design Tendencies. Figure 12 illustrates exemplary vibrotactile patterns designed using HapticPilot. It is interesting to note the variety of design strategies using HapticPilot. For example, some participants focused on drawing haptic traces within the hand region for different postures (*Action*, 'Attack', *Finger Gun*) to concentrate vibrotactile sensation, whereas other participants utilized the whole drawable area to deliver shape information (*Puzzle*, '6', *Thumbs Up*).

6 APPLICATIONS

Our posture-adaptive algorithm is capable of preserving directional sensation and helping users distinguish between within- and outside-palm haptic points. In this section, we highlight specific use case scenarios using glove-type haptic devices where HapticPilot could be exceptionally helpful. Here, the highlighted applications require abstract and creative haptic designs with directional context rather than drawing specific alphabets or digits.

VR Immersion Improvement Nowadays, users utilize controllers to carry out haptic design tasks for VR VFX, where users' hand postures are often restricted. Although various haptic gloves have been introduced, it is difficult to design haptic experiences for these devices because of unpredictable hand postures. HapticPilot

Fig. 13. Potential haptic design scenarios with HapticPilot. The user designs (a) VR VFX for weapons with different hand postures while receiving the same haptic experience, (b) uniform haptic notification for incoming mail regardless of the user's hand postures, and (c) haptic guidance with consistent direction sensation for tools with different hand grasps.

ensures the same haptic experience across various hand postures, ultimately improving VR immersion with haptic gloves for game scenes (Figure 13a).

Effective Notifications For common notifications like calls or texts in VR, haptic design should provide a balanced sensation to provide effective interruption while minimizing irritation from users. Utilizing HapticPilot's in-situ authoring, users could easily iterate the design drafts to provide balanced sensations over various hand postures. Moreover, HapticPilot provides adequate design tools for users to create distinctive haptic designs for different notifications in each context. Figure 13b illustrates how HapticPilot can help design adequate notifications in a virtual office environment.

Enhanced Guidance & Training HapticPilot preserves the stimulus's directional sensation on various hand postures, which allows such information to be transmitted accurately. This can be mainly used for vibrotactile navigation in VR [9] such as maps or complex training simulations used for aviation or surgery, which require users to employ different hand postures. Figure 13c shows an example of HapticPilot designs used for navigation in a training scenario. The above-mentioned scenarios illustrate a glimpse of potential applications using HapticPilot, which are selected to highlight the results of our studies. HapticPilot could also be employed for any other applications that require designing whole-hand vibrotactile sensations on various hand postures with minimal information loss.

7 DISCUSSION

In this section, we interpret the results and share our observations from user studies. Furthermore, we discuss the current limitations of the proposed authoring toolkit and suggestions for future iterations.

Information Transfer vs. Design Intention HapticPilot showed superior performance in rendering digits and alphabets compared to shapes. In Study 1, our algorithm scored higher accuracy in the digit category compared to the shape category. For Study 2, the alphabet scored the highest similarity score (5.6) among all. However, users rated the usability of *Action* scenes higher than *Puzzle* scenes which used familiar alphabets, symbols, and digits in Study 2. We assume this is due to different writing styles of alphabets and digits for people with various backgrounds. For creative patterns used in *Action* scenes, users were generally satisfied as long as the user "feels" the design intention. This also tells that users would set higher standards for vibrotactile feedback conveying preconceived shape information like alphabets and digits.

Towards Personalized Haptic Design During the usability evaluation, we observed that participants preferred their own designs (Q2-2: AVG=5.3, SD=2.0) over designs made by others (Q2-1: AVG=4.3, SD=1.6). From the result, we infer that users prefer the personalized and customized haptic design over the general vibrotactile patterns. In this regard, we believe that a user-friendly haptic design toolkit with a lower workload, like HapticPilot, would benefit not only existing hapticians but also any user who likes to design his or her own vibrotactile patterns.

High Standards for Authoring Interface During the qualitative feedback, HapticPilot received favorable scores ($\geq$ 5.8) for interface-related areas (Q3~Q5). We assume positive feedback came from adopting a familiar interface design (button-based 3D UI) and interactions (pen-based sketching). However, participants tend to compare our interface with higher standards such as commercially available VR drawing applications. Thus, we believe the current usability score is at the high end as a research prototype considering users' high standards.

Limitations and Future Works The current vision technology used in industrial VR HMDs for hand tracking is not yet problem-free, shown by occlusion when hands are overlapped. Despite our attempts at user interaction such as grabbing haptic intensity pens, this drawback negatively impacted system usability. We could address this issue by enhancing hand-tracking methods or by developing user interaction with alternative modalities. Employing haptic gloves with embedded physical sensors could be an alternative to vision-based

hand tracking [26]. Moreover, incorporating alternative input modalities such as audio, text, and visuals could act as an effective strategy to avoid the occlusion.

In this work, the toolkit was designed to support predefined representative hand postures only on the left hand. However, the same algorithm could be mirrored to the right hand to support bimanual haptic design. A future approach would be utilizing the hand posture-capturing functions from commercial SDKs to add and modify users' hand postures freely. Furthermore, we plan to provide HapticPilot as plug-ins that could be integrated into real game development studios. Since the algorithm was based on a 2D phantom sensation method, the haptic sketches were drawn in a 2D plane on hand. Also, the supported hand postures were restricted to those whose fingers were either entirely extended or flexed to touch the palm. By integrating other methods such as out-of-the-body sensation [9, 33, 41], however, we expect to expand the rendering domain into 3D spaces.

8 CONCLUSION

HapticPilot is an in-situ VR haptic authoring framework that utilizes a posture-adaptive haptic rendering algorithm to address the need for a posture-adaptive algorithm. We devised phantom grid which is a novel haptic design abstraction that represents spatial information of the hand structure. With the phantom grid, our algorithm enables rendering intended haptic feedback across various hand postures. We further proposed a unique whole-hand actuator arrangement by integrating 1D & 2D phantom sensations. In the haptic experience evaluation, the proposed posture-adaptive rendering algorithm achieved comparable IT, accuracy, and similarity scores with 20 participants. For system usability evaluation, we collected both qualitative feedback and an overall SUS score of 81 with another 20 participants. The in-depth interviews indicated that HapticPilot's design through sketching and hand posture-related features are efficient and user-friendly. Based on these observations, we demonstrated application scenarios where our system enhances VR immersion. We believe that HapticPilot will open up a myriad of possibilities by enabling both experienced and novice hapticians to create personalized, robust, and rich haptic designs that are applicable to any hand posture.

ACKNOWLEDGEMENTS

This work was supported by the National Research Council of Science & Technology (NST) (No. CRC21011) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 00210001 & No. 2022R1A4A5033689).

REFERENCES

[1] David S Alles. 1970. Information transmission by phantom sensations. *IEEE transactions on man-machine systems* 11, 1 (1970), 85–91.

[2] Leia B Bagesteiro and Robert L Sainburg. 2003. Nondominant arm advantages in load compensation during rapid elbow joint movements. *Journal of neurophysiology* 90, 3 (2003), 1503–1513.

[3] bHaptics. 2023. TactGloves, haptic gloves for VR. https://www.bhaptics.com.

[4] Fabien Danieau, Philippe Guillotel, Olivier Dumas, Thomas Lopez, Bertrand Leroy, and Nicolas Mollet. 2018. HFX studio: haptic editor for full-body immersive experiences. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. 1–9.

[5] Donald Degraen, Bruno Fruchard, Frederik Smolders, Emmanouil Potetsianakis, Seref Güngör, Antonio Krüger, and Jürgen Steimle. 2021. Weirding Haptics: In-Situ Prototyping of Vibrotactile Feedback in Virtual Reality through Vocalization. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 936–953.

[6] Diane Dewez, Ludovic Hoyet, Anatole Lécuyer, and Ferran Argelaguet Sanz. 2021. Towards "avatar-friendly" 3D manipulation techniques: Bridging the gap between sense of embodiment and interaction in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[7] Ulrike Gollner, Tom Bieling, and Gesche Joost. 2012. Mobile lorm glove: introducing a communication device for deaf-blind people. In *Proceedings of the sixth international conference on tangible, embedded and embodied interaction*. 127–130.

[8] Xiaochi Gu, Yifei Zhang, Weize Sun, Yuanzhe Bian, Dao Zhou, and Per Ola Kristensson. 2016. Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in VR. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1991–1995.

[9] Sebastian Günther, Florian Müller, Markus Funk, Jan Kirchner, Niloofar Dezfuli, and Max Mühlhäuser. 2018. Tactileglove: Assistive spatial guidance in 3d space through vibrotactile navigation. In *Proceedings of the 11th pervasive technologies related to assistive environments conference*. 273–280.

[10] Mark Hollins, Sliman Bensmaïa, Kristie Karlof, and Forrest Young. 2000. Individual differences in perceptual space for tactile textures: Evidence from multidimensional scaling. *Perception & Psychophysics* 62 (2000), 1534–1544.

[11] HTC. 2023. VIVE Hand Tracking SDK. Retrieved May 16, 2023 from https://developer.vive.com/resources/vive-sense/hand-tracking-sdk.

[12] Da-Yuan Huang, Liwei Chan, Xiao-Feng Jian, Chiun-Yao Chang, Mu-Hsuan Chen, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. Vibroplay: Authoring three-dimensional spatial-temporal tactile effects with direct manipulation. In *SIGGRAPH ASIA 2016 Emerging Technologies*. 1–2.

[13] HaptX Inc. 2022. Haptic gloves G1. https://haptx.com/.

[14] Ali Israr and Ivan Poupyrev. 2011. Tactile brush: drawing on skin with a tactile grid display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2019–2028.

[15] Roland S Johansson and J Randall Flanagan. 2009. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience* 10, 5 (2009), 345–359.

[16] Topi Kaaresoja, Stephen Brewster, and Vuokko Lantz. 2014. Towards the temporally perfect virtual button: touch-feedback simultaneity and perceived quality in mobile touchscreen press interactions. *ACM Transactions on Applied Perception (TAP)* 11, 2 (2014), 1–25.

[17] Idin Karuei, Karon E MacLean, Zoltan Foley-Fisher, Russell MacKenzie, Sebastian Koch, and Mohamed El-Zohairy. 2011. Detecting vibrations across the body in mobile contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 3267–3276.

[18] Oliver Beren Kaul, Andreas Domin, Michael Rohs, Benjamin Simon, and Maximilian Schrapel. 2021. VRTactileDraw: A Virtual Reality Tactile Pattern Designer for Complex Spatial Arrangements of Actuators. In *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part V*. Springer, 212–233.

[19] Oliver Beren Kaul and Michael Rohs. 2017. HapticHead: A Spherical Vibrotactile Grid around the Head for 3D Guidance in Virtual and Augmented Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3729–3740. https://doi.org/10.1145/3025453.3025684

[20] Erin Kim and Oliver Schneider. 2020. Defining haptic experience: foundations for understanding, communicating, and evaluating HX. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[21] Jinsoo Kim, Seungjae Oh, Chaeyong Park, and Seungmoon Choi. 2020. Body-Penetrating Tactile Phantom Sensations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376619

[22] Jaeha Kim, Yonghwan Oh, and Jaeyoung Park. 2017. Adaptive vibrotactile flow rendering of 2.5D surface features on touch screen with multiple fingertip interfaces. In *2017 IEEE World Haptics Conference (WHC)*. 316–321. https://doi.org/10.1109/WHC.2017.7989921

[23] Taejun Kim, Youngbo Aram Shim, and Geehyuk Lee. 2021. Heterogeneous stroke: Using unique vibration cues to improve the wrist-worn spatiotemporal tactile display. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[24] Kishor Lakshminarayanan, Abigail W Lauer, Viswanathan Ramakrishnan, John G Webster, and Na Jin Seo. 2015. Application of vibration to wrist and hand skin affects fingertip tactile sensation. *Physiological reports* 3, 7 (2015), e12465.

[25] Jaeyeon Lee, Mike Sinclair, Mar Gonzalez-Franco, Eyal Ofek, and Christian Holz. 2019. TORC: A virtual reality controller for in-hand high-dexterity finger interaction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[26] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking Using EMG Wearables. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1471–1482. https://doi.org/10.1145/3442381.3449890

[27] Matthew R Longo. 2017. Hand posture modulates perceived tactile distance. *Scientific Reports* 7, 1 (2017), 9665.

[28] Hu Luo, Zemin Wang, Zhicheng Wang, Yuru Zhang, and Dangxiao Wang. 2023. Perceptual Localization Performance of the Whole Hand Vibrotactile Funneling Illusion. *IEEE Transactions on Haptics* (2023).

[29] Louise R Manfredi, Andrew T Baker, Damian O Elias, John F Dammann III, Mark C Zielinski, Vicky S Polashock, and Sliman J Bensmaia. 2012. The effect of surface wave propagation on neural responses to vibration in primate glabrous skin. *PloS one* 7, 2 (2012), e31203.

[30] Jonatan Martínez, Arturo García, Miguel Oliver, José Pascual Molina, and Pascual González. 2014. Identifying virtual 3D geometric shapes with a vibrotactile glove. *IEEE computer graphics and applications* 36, 1 (2014), 42–51.

[31] Meta. 2022. Design for Hands. Retrieved May 16, 2023 from https://developer.oculus.com/resources/hands-design-interactions.

[32] Microsoft. 2022. Hand menu. Retrieved May 16, 2023 from https://learn.microsoft.com/en-us/windows/mixed-reality/design/hand-menu.

[33] Makoto Miyazaki, Masaya Hirashima, and Daichi Nozaki. 2010. The "cutaneous rabbit" hopping out of the body. *Journal of Neuroscience* 30, 5 (2010), 1856–1860.

[34] Makiko Natsume, Yoshihiro Tanaka, Wouter M Bergmann Tiest, and Astrid ML Kappers. 2017. Skin vibration and contact force in active perception for roughness ratings. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 1479–1484.

[35] Oculus. 2022. Oculus Integration SDK. https://developer.oculus.com/downloads/package/unity-integration.

[36] Oliver Ozioko and Ravinder Dahiya. 2022. Smart tactile gloves for haptic interaction, communication, and rehabilitation. *Advanced Intelligent Systems* 4, 2 (2022), 2100091.
[37] Gunhyuk Park, Hojun Cha, and Seungmoon Choi. 2018. Haptic enchanters: Attachable and detachable vibrotactile modules and their advantages. *IEEE transactions on haptics* 12, 1 (2018), 43–55.
[38] Gunhyuk Park and Seungmoon Choi. 2018. Tactile information transmission by 2D stationary phantom sensations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.
[39] Jaeyoung Park, Jaeha Kim, Yonghwan Oh, and Hong Z Tan. 2016. Rendering moving tactile stroke on the palm using a sparse 2d array. In *Haptics: Perception, Devices, Control, and Applications: 10th International Conference, EuroHaptics 2016, London, UK, July 4-7, 2016, Proceedings, Part I 10*. Springer, 47–56.
[40] Dario Pittera, Elia Gatti, and Marianna Obrist. 2019. I'm sensing in the rain: Spatial incongruity in visual-tactile mid-air stimulation can elicit ownership in VR users. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.
[41] Dario Pittera, Marianna Obrist, and Ali Israr. 2017. Hand-to-hand: an intermanual illusion of movement. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 73–81.
[42] Dongseok Ryu, Gi-Hun Yang, and Sungchul Kang. 2009. T-hive: Vibrotactile interface presenting spatial information on handle surface. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, 683–688.
[43] Ioannis Sarakoglou, Anais Brygo, Dario Mazzanti, Nadia Garcia Hernandez, Darwin G. Caldwell, and Nikos G. Tsagarakis. 2016. HEXOTRAC: A highly under-actuated hand exoskeleton for finger tracking and force feedback. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1033–1040. https://doi.org/10.1109/IROS.2016.7759176
[44] Oliver Schneider, Karon MacLean, Colin Swindells, and Kellogg Booth. 2017. Haptic experience design: What hapticians do and where they need help. *International Journal of Human-Computer Studies* 107 (2017), 5–21.
[45] Oliver S Schneider, Ali Israr, and Karon E MacLean. 2015. Tactile animation by direct manipulation of grid displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 21–30.
[46] Oliver S Schneider and Karon E MacLean. 2016. Studying design process and example use with Macaron, a web-based vibrotactile effect editor. In *2016 IEEE Haptics Symposium (HAPTICS)*. IEEE, 52–58.
[47] Hasti Seifi, Matthew Chun, Colin Gallacher, Oliver Schneider, and Karon E MacLean. 2020. How do novice hapticians design? A case study in creating haptic learning environments. *IEEE transactions on haptics* 13, 4 (2020), 791–805.
[48] BeBop Sensors. 2019. Forte Glove, AR/VR gloves. https://bebopsensors.com.
[49] Yitian Shao. 2022. *Tactile Sensing, Information, and Feedback Via Wave Propagation*. Springer.
[50] Bukun Son and Jaeyoung Park. 2018. Tactile sensitivity to distributed patterns in a palm. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*. 486–491.
[51] Miru Studio. 2022. Finger Gun. Retrieved May 16, 2023 from https://mirustudio.eu/finger-gun.
[52] Hong Z Tan, Charlotte M Reed, and Nathaniel I Durlach. 2009. Optimum information transfer rates for communication through haptic and other sensory modalities. *IEEE Transactions on Haptics* 3, 2 (2009), 98–108.
[53] Velko Vechev, Juan Zarate, David Lindlbauer, Ronan Hinchet, Herbert Shea, and Otmar Hilliges. 2019. Tactiles: Dual-mode low-power electromagnetic actuators for rendering continuous contact and spatial haptic patterns in VR. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 312–320.
[54] Ana Villanueva, Zhengzhe Zhu, Ziyi Liu, Feiyang Wang, Subramanian Chidambaram, and Karthik Ramani. 2022. Colabar: A toolkit for remote collaboration in tangible augmented reality laboratories. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–22.
[55] Yon Visell and Yitian Shao. 2016. Learning constituent parts of touch stimuli from whole hand vibrations. In *2016 IEEE Haptics Symposium (HAPTICS)*. IEEE, 253–258.
[56] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 552–567.
[57] Jacob O Wobbrock, Brad A Myers, and John A Kembel. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. 61–70.
[58] Chien-Min Wu, Chih-Wen Hsu, Tzu-Kuei Lee, and Shana Smith. 2017. A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. *Virtual Reality* 21 (2017), 19–29.
[59] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. 2018. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[60] Youngwoo Yoon, Keunwoo Park, Minsu Jang, Jaehong Kim, and Geehyuk Lee. 2021. Sgtoolkit: An interactive gesture authoring toolkit for embodied conversational agents. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 826–840.
[61] Massimiliano Zampini, Charlotte Harris, and Charles Spence. 2005. Effect of posture change on tactile perception: impaired direction discrimination performance with interleaved fingers. *Experimental Brain Research* 166 (2005), 498–508.

A EXPLORATORY STUDY OF WHOLE-HAND SKIN VIBRATIONS: HEATMAP VISUALIZATION

Fig 14 shows the entire skin vibration heatmaps obtained through exploratory study 1. The heatmaps for the *Rest* posture (large heatmaps) and the other postures (small heatmaps) are illustrated with the motor region indicated.

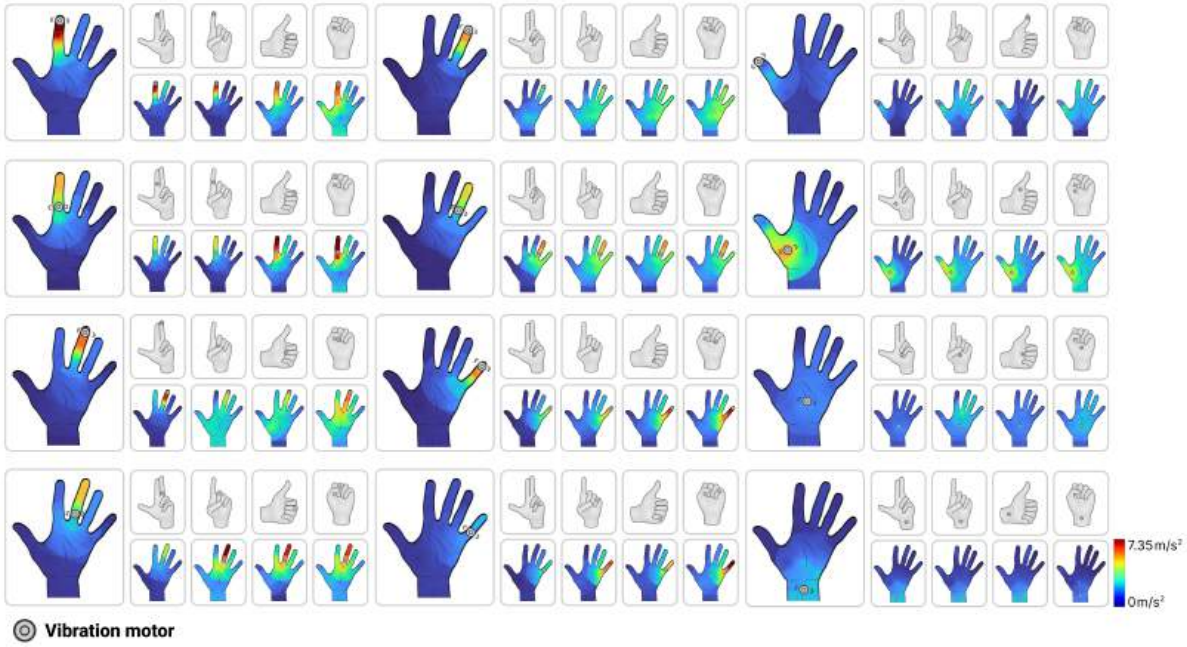

Fig. 14. The whole-hand skin vibrations elicited from 12 motor regions on five hand postures.

B ALGORITHMS

Algorithm 1 describes the process used to create the phantom grid. Algorithm 2 shows the posture-adaptive rendering algorithm. Algorithm 3 and 4 are helper functions used to render haptic points that need to be mapped to the nearest point during the posture-adaptive rendering algorithm.

Algorithm 1: Phantom Grid construction algorithm

Input : *nodeSet* is the set of all haptic nodes N on the hand with each haptic node containing information of its *origin*. By default, the *contact* of a haptic node is the same as the *origin*.

Output: *phantomGrid* is a class object that contains several information about the phantom grid.

1. Label the motors ;

foreach $N \in nodeSet$ **do**
  $N.contact = LabelNodes(N)$ ▷ *detect where the haptic node contacts*
$palmContactNodes := \{n | n.contact = palm\}$

2. Analyze the haptic node relations ;

Function *ConstructPhantomGrid*():
  foreach $finger \in fingers$ **do** `// Finger-finger 1D`
    two consecutive nodes on a finger create a ***grid line***
  foreach $finger \in flexionFingers$ **do** `// Finger-palm 1D`
    lowest node in the finger and palm node create a ***grid line***
  foreach $n1 \in palmContactNodes$ **do** `// Palm-palm 1D`
    foreach $n2 \in palmContactNodes$ **do**
      if $distance(n1, n2) < threshold$ **then**
        $(n1, n2)$ creates a ***grid line***
  palmContactNodes creates a ***palm boundary*** `// Palm-palm 2D`
  2.2 Hand boundary
  create a concave hull(***hand boundary***) consisted of the following motors
    (1) *outermost node of each finger*
    (2) *nodes of the leftmost and rightmost flexion fingers*
    (3) *leftmost, rightmost, and bottommost nodes of the palm*
  return

Algorithm 2: Flow of rendering a haptic point

Inputs : P is the haptic point to render, A is the target haptic amplitude
Outputs : $motorAmpList$ is a list of three tuples (M_i, A_i) where $1 \leq i \leq 3$. M_i is the motor and A_i is the amplitude of the specific motor

Function $motorAmpList = RenderPattern(P, A)$**:**

- **foreach** *motor* $M \in motorSet$ **do** `// Haptic node collision`
  - **if** $Distance(M, P) \leq motorSize$ **then**
    - **return** $[(M, A), (0, 0), (0, 0)]$;
- **foreach** *line* $L \in gridLinesSet$ **do**
  - **if** $Distance(L, P) \leq lineWidth$ **then** `// Grid line intersection`
    - $M_1, M_2 \leftarrow L.consistMotors$;
    - $A_1, A_2 \leftarrow 1DPhantom(M_1, M_2, P, A)$;
    - **return** $[(M_1, A_1), (M_2, A_2), (0, 0)]$;
- `// Use ray casting algorithm for polygon`
- $C \leftarrow$ Haptic node located at the center of palm;
- $direction = C - P$;
- $hits = Raycast(P, direction)$;
- $n \leftarrow 0$;
- **foreach** *hit* $\in hitLines$ **do**
  - **if** $hit.line \in phantomGrid.palmBoundary$ **then**
    - $n++$;
- **if** $n\%2 == 1$ **then** `// Within-palm localization`
  - $triangle \leftarrow FindTriangleEnclosing(P)$;
  - $M_1, M_2, M_3 \leftarrow triangle.consistMotors$;
  - $A_1, A_2, A_3 \leftarrow 2DPhantom(M_1, M_2, M_3, P, A)$;
  - **return** $[(M_1, A_1), (M_2, A_2), (M_3, A_3)]$;
- **else** `// Outside-palm localization`
  - $P' = MapNearestPoint(P)$;
  - $A' = 0.1 * A$;
  - **return** $RenderPattern(P', A')$;

Algorithm 3: Nearest point mapping

Input : P is the target point
Output: Nearest point $P\prime$ on the $phantomGrid$
Function $P' = MapNearestPoint(P)$**:**

```
C ← Haptic node located at the center of palm;
direction = C − P;
hitPoint1, hitPoint2 ← (0, 0, 0);
hitLine1, hitLine2 ← null;
hitsTowards ← Raycast(P, direction);
foreach hit ∈ hitTowards do
    if hit.line ∈ phantomGrid.handBoundary then
        hitPoint1 = hit.point;
        hitLine1 = hit.line;
        break;
hitsAway ← Raycast(P, −direction);
foreach hit ∈ hitsAway do
    if hit.line ∈ phantomGrid.handBoundary then
        hitPoint2 = hit.point;
        hitLine2 = hit.line;
        break;
if hitLine1 == null then
    return CheckImaginary(hitLine2, hitPoint2);
if hitLine2 == null then
    return CheckImaginary(hitLine1, hitPoint1);
distance1 ← distance(P, hitLine1.motor1) + distance(P, hitLine1.motor2);
distance2 ← distance(P, hitLine2.motor1) + distance(P, hitLine2.motor2);
if distance1 < distance2 then
    return CheckImaginary(hitLine1, hitPoint1);
else
    return CheckImaginary(hitLine2, hitPoint2);
```

Algorithm 4: Algorithm to check if the point is mapped to an imaginary line

```
Input  : Line L and point P
Output : New point P'
Function P' = CheckImaginary(L,P):
    if L.isValid then
        return P;
    distance1 ← distance(P, L.motor1);
    distance2 ← distance(P, L.motor2);
    if distance1 < distance2 then
        return L.motor1;
    else
        return L.motor2;
```