HapticSphere: Physical Support To Enable Precision Touch Interaction in Mobile Mixed-Reality

Chiu-Hsuan Wang Chen-Yuan Hsieh Neng-Hao Yu† Andrea Bianchi‡ Liwei Chan
† Department of Industrial Design, NTUST, Taiwan
‡ Department of Industrial Design, KAIST, Korea
Department of Computer Science, National Chiao Tung University, Taiwan

Abstract

This work presents HapticSphere, a wearable spherical surface enabled by bridging a finger and the head-mounted display (HMD) with a passive string. Users perceive a physical support on a finger attached to a string, when extending their arm and reaching out to the string's maximum extension. This physical support assists users in precise touch interaction in the context of stationary and walking virtual or mixed-reality experiences. We propose three methods of attachment of the haptic string (directly on the head or on the body), and illustrate a novel single-step calibration algorithm that supports these configurations by estimating a grand haptic sphere, once a head-coordinated touch interaction is established. Two user studies were conducted to validate our approach and to compare the touch performance with physical support in sitting and walking conditions in the context of mobile mixed-reality scenarios. The results show that, in the walking condition, touch interaction with physical support significantly outperformed the visual-only condition.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

Touchscreens provide physical support for fingers during touch interaction, implicitly helping users with haptic feedback and a form of finger stabilization in the case of input on the go. Unfortunately, current mixed-reality interaction that employs in-air touch input does not offer such physical support, which may lead to lowered performance. Recent research on virtual reality has demonstrated a range of active haptic interfaces integrated into wearable devices [13,16] or hand-held controllers [9,22] to deliver force feedback directly onto the users' fingers. They allow for *locally bound* physical support at the finger but are incapable of limiting the arm's motion, providing only a localized kinesthetic feedback or tactile sensation.

This work presents *HapticSphere*, a system that integrates a finger tracking device with a passive string attached to an HMD, to provide physical support at the user's finger for midair touch interaction. The envisioned application domain is mobile mixed-reality interactions. We acquire physical support by linking the user's finger to his or her body with a passive string that works as a constraint for the finger's motion. Figure 1 illustrates an example of touch interaction constrained by the *HapticSphere* prototype adding to the user's head (e.g., the HMD). The user perceives physical support at the finger when reaching the maximum extension of the string, at which moment the physical support provides both input stabilization and haptic guidance for touch interaction. This support has the shape of a spherical wall surrounding the user, hence the name *HapticSphere*. This force feedback assists users during mixed-reality experiences for precise touch interaction in both static and mobile situations (e.g., walking), and it also informs the user that he or she has reached and clicked the in-air target.

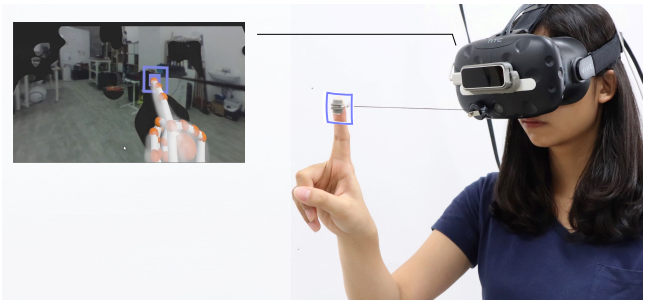

Figure 1: The physical support enabled by *HapticSphere* allows users to perform a precise touch input using in-air selections during mobile mixed-reality explorations.

Compared with previous work that employed flexible passive strings to constrain the full motion of an entire arm [2,3], in this paper we focus on providing force feedback on the fingertip for precise input interaction. Our ultimate goal is to support precise touch-input selections in mixed-reality mobile situations. To achieve this goal, we present the *HapticSphere* system and two studies that demonstrate its accuracy during selections of in-air targets. Specifically, we propose and study a novel single-step calibration procedure to acquire the *grand haptic sphere*, which is adaptable to various wearable conditions of the string on the user's body. We compare three ways in which to bind the string to the user's body (e.g., by HMD, by neck, and by shoulder), and we investigate how physical support assists touch accuracy for different sized targets and in both sitting and walking conditions. Study 1 demonstrated the effectiveness of the *grand haptic sphere* algorithm and revealed how physical support assists in-air target acquisition in reducing overshooting errors. Study 2 reported that, in walking conditions, touch interaction with physical support significantly outperforms the visual-only condition in terms of touch accuracy. Our main contribution is the idea of a *wearable* spherical surface that allows for physical support for in-air target acquisition in the context of mobile mixed-reality.

2 Related Work

We review work that enabled wearable force (kinesthetic) feedback on the hand/finger for mobile virtual and mixed-reality applications, as well as string-based haptic interactions.

2.1 Wearable Force Feedback Interfaces

Bringing force feedback to virtual reality allows for leveling up immersion and boosting user performance. Recent research proposed a range of means of adding force feedback on the user's limbs [18,24],

e-mail: [chwang821014, liweichan]@cs.nctu.edu.tw
†e-mail: jonesyu@ntust.edu.tw
‡e-mail: andrea@kaist.ac.kr

2019 IEEE Conference on Virtual Reality and
3D User Interfaces
23-27 March, Osaka, Japan
978-1-7281-1377-7/19/$31.00 ©2019 IEEE

head [17], face [11], and body [14]. In this work, we focus on previous work on finger- and hand-scale force feedbacks.

In finger-scale feedback, Gravity Grabber [26] simulates the weight of a virtual object in a grip with a shear force applied to the fingers. Traxion [27] enables the illusion of a dragging force on the user's fingers through a tactile actuator injecting asymmetric signals. HapThimble [22] enables tactile and pseudo-force feedback on a mechanically augmented finger, which simulates the sensation of pressing a physical button. Benko et. al. [9] renders a virtual shape by actuating the user's finger resting on the hand controller's built-in motion platform.

Hand-scale force feedback is primarily achieved via handheld controllers and wearable interfaces. Handheld controllers are augmented to generate force feedback using the mechanics of the gyro effect [30], wind propulsion [18, 21], and shifting weights [28, 32]. Haptic Link [29] leverages a mechanical link between controllers for bimanual force feedback. In addition, early work developed glove-based exoskeletons [1] employing complex mechanisms installed around the hand, which could add weight and limit the interaction space. More recently, Dexmo [16], Wolverine [13], and DextrES [19] presented lightweight braking exoskeleton interfaces that produce resistive forces on multiple fingers and enable users to physically grasp virtual objects. Pneumatic proxy [31] is employed to simulate object shapes in grasps, and electrical muscle stimulation [24] is used to generate an impact force. In addition to active haptics, Virtual Mitten [4] presents a passive elastic hand grip, but its applicability is limited to gripping interactions. All interfaces described above provide localized force feedback on the user's fingertip or hands, but they are limited by not being able to constrain the rest of the body or the arms' motions.

2.2 String-Based Haptic Interactions

String-based interfaces can both limit or assist wider motions involving the arms or other body parts. SPIDAR [5] is the first example of a string-based haptic interface for virtual reality applications. It connects multiple strings to the user's fingertips, whereas the underlying driving mechanism allows for simultaneously measuring the fingers' positions and to rendering force on them. Unlike SPIDAR's active haptics, Paljic et al. [6] developed a passive stringed system with a user-actuated brake mechanism to display a stiffness sensation. The abovementioned string systems are deployed in the environment, which constrains the interaction space and the mobility of the users.

To enable mobile interaction, SPIDAR-S [25] reduces the previous SPIDER system to a single active string device that deploys dynamic force feedback on a single finger. HapticGear [20] presents a backpacked active string system that displays active force feedback to a pen connected to a string. The string system provides physical support to the pen interactions, thus reducing fatigue during writing. ElasticArm [2,3] presents a body-mounted elastic armature that links the users palm to his or her shoulder, which creates a progressive resistance force on the palm when the user extends his or her arm. Like ElasticArm, *HapticSphere* is also a passive single-string interface, but unlike ElasticArm's progressive displacement of force on the palm, we designed *HapticSphere* to produce a solid force feedback on the fingertip. In addition, in our work, we aim to explore how physical support helps mobile touch interactions for mixed-reality and mobile targeting tasks.

3 System Implementation

HapticSphere's hardware components consist of a mobile HMD capable of three-dimensional (3D) finger tracking, and a piece of passive string that connects the user's finger to the user's body through the HMD or other accessories.

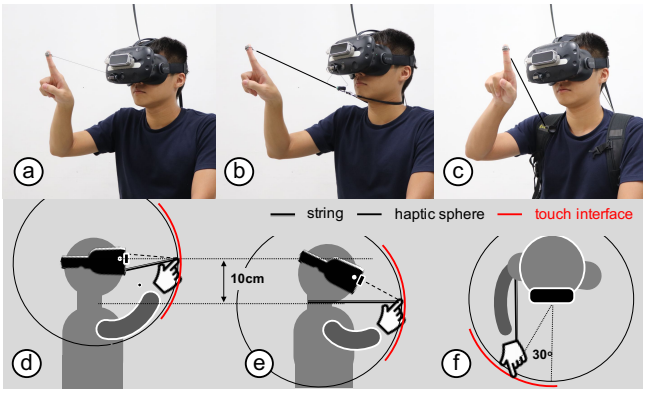

Figure 2: Three attachment methods: attaching(a) by the HMD, (b) by neck as a neck strap, and (c) by the shoulder through a backpack strap. (d)(e)(f) their interaction spaces respectively.

3.1 Finger Input

Although the next commercial HMDs may be equipped with a finger tracking function, this feature is currently not available on the HMD of our prototype. For a proof-of-concept prototype, we implemented the system using a HTC Vive headset enhanced with a LeapMotion[1] finger tracker. As shown in Figure 1, the LeapMotion tracker is located at the front face of the headset for hand tracking. The software was implemented using Unity3D[2] engine. LeapMotion plugin for Unity3D allowed us to access reliable finger tracking data.

3.2 String Attachments

Figure 2 illustrates three different ways in which to attach the string to the user's body: direct attachment (1) to the HMD device, (2) to the neck through a neck strap, or (3) to the shoulder through a backpack strap. In the rest of the paper, we refer to them as *by-HMD*, *by-neck*, and *by-shoulder* attachments. The three methods primarily differ in how the users perceive the force on the finger, and by the interaction space that each method allows.

To keep the string from interfering with the LeapMotion's tracking performance, we used a thin string reel modified from a typical retractable reel. We disabled the spring mechanic inside the reel to remove the resistive force that might cause finger fatigue during interactions. This means finger motion is unconstrained by the string until the finger reaches the maximum string extension. At such a point, the string acts as a stopper. To provide a grip for the finger, we tied a finger pad at the end of the string.

In the *by-HMD* condition, the reel is attached below the LeapMotion so as not to occlude the camera views. In the *by-neck* condition, the reel is worn as a neck strap, and in the *by-shoulder* condition, it is worn with a backpack strap. The string length is customized for each of the three attachment methods to keep the same touch distance of a target from the user: their lengths are 27 cm in the *by-HMD* and 29 cm in the *by-neck* and *by-shoulder* arrangements. Although our implementation decided to remove the reel's spring mechanic, an alternative design can choose to retrofit the retractable string for the ease of the disengagement of the interface from the finger, which we will discuss in the Discussion section.

3.2.1 Force Perception

When users pull the string to its extent, both the finger and the part of the body on which the string is attached perceive the same amount of force in opposite directions. The force at the finger is the desired

[1]LeapMotion: https://www.leapmotion.com/
[2]Unity3D: https://unity3d.com/

feedback, whereas the force perceived on the other parts of the body (head, neck, shoulder) is a by-product of the interaction. In the *by-HMD* interface, the force at the finger feels sharper and more responsive than that in the *by-neck* and *by-shoulder* interfaces. This is because the solid body of the HMD provides a direct force transfer through the string, compared with a transfer that goes through the soft surface of the neck in the *by-neck* interface and the backpack strap in the *by-shoulder* interface. In the latter two configurations, the string behaves as a spring that slows the transmission and softens the perceived force on the finger. On the contrary, users perceive a direct, instantaneous counterforce on the head through the HMD in the *by-HMD* interface, a soft pressing force on the back of the neck in the *by-neck* interface, and a negligible force on the shoulder in the *by-shoulder* interface.

3.2.2 Interaction Space

The string defines an interaction space, centering its attachment point on the user's body, and constraining the arm motion in the space enclosed by a haptic sphere. Figure 2 illustrates the interaction space for each attachment methods. In the *by-HMD* and *by-neck* interfaces, the interaction spaces are generally in front of the HMD, with that of the *by-neck* being 10 cm below. In the *by-shoulder* interface, the interaction space is set at the same height as that of the *by-neck* interface, and it is turned toward the side of the backpack strap by 30 degrees. Catering the modifications to each attachment methods suggests better ergonomics during touch interaction. These modifications were adopted in the Study 1.

4 Enabling Spherical Physical Support

The passive string attached to the user's body creates a wearable haptic sphere that always follows the user. By locating the haptic sphere in the virtual space using the HMD as a reference point, we enable haptic touch interaction on the sphere's inner surface.

4.1 Estimation of a HapticSphere

Here we describe a simple calibration procedure to locate the haptic sphere with a finger-tracking-ready HMD. This requires the user to provide more than four touch points on the surface of the haptic sphere, which can be easily collected by having the finger write at the maximum extension of the string. In our implementation, users are instructed to draw a long continuous stroke following a grid that completely covers the HMD view (Figure 6). On the stroke, we collected 1000 touch points and approximated an sphere using the random sample consensus algorithm [15], which is less subject to outliers compared with least squares fitting, thus being unaffected by LeapMotion's occasional tracking failures. Intuitively, a visual touch interface would be laid on the haptic sphere.

4.2 Analysis in Visual-Haptic Alignment

A haptic sphere formed by three attachments is either tight-coupled or loose-decoupled with the HMD. In the tight-coupled case (e.g., by-HMD), the haptic sphere is constrained to the HMD by the string extent. In the viewpoint of the HMD, the haptic sphere is fixed and unique (Figure 3ab), regardless of the user's heading. Thus, single calibration is sufficient for determining a haptic sphere for a touch interface.

In loosely coupled cases (e.g., *by-neck* and *by-shoulder*), the string is a separate component of the HMD and thus is not directly influenced by the HMD movement. Because the string now rotates using the neck or shoulder, instead of the HMD, a haptic sphere in the viewpoint of the HMD would shift, albeit slightly, in response to the user's heading (Figure 3cd). This shift is attributed to the relative motion between the HMD and the attachment points. To ensure coherent visuo-haptic sensation, one needs to capture this shift and apply it back to the haptic sphere. The same situation applies to the *by-shoulder* interface. However, in the case of a *by-HMD* interface, the shift is constantly zero.

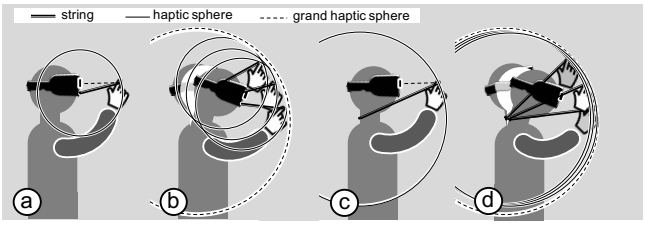

Figure 3: (a)(c) a sample haptic sphere for HMD and By-Neck attachments when users fact the front. (b)(d) The haptic sphere moves with the neck rotation, and *Grand haptic sphere* is an estimation of the union of all haptic spheres.

Figure 4: Head-coordinated touch interaction in the case of by-HMD interface. The strings in the photos were retouched for visibility.

4.3 Haptic Sphere Revisited

Instead of capturing relative motions between the attachment point and the HMD, our new approach works for all three attachments by estimating a *grand haptic sphere* intended to collect the integral behaviors of a user performing a *head-coordinated touch* interaction at the string maximum extension. With the phrase *head coordinated touch*, we mean that the user would look at a target through the viewport center (e.g., aiming at the target with the heading) while touching the target. Figure 4 shows the user's motions, whereas a head-coordinated touch is made with the *by-HMD* interface.

Technically speaking, the *grand haptic sphere* captures the union of all of the possible haptic spheres in association with the attachment point in the viewpoint of the HMD. Figure 3bd shows the *grand haptic sphere* of the *by-HMD* and of the *by-neck* interfaces, respectively. With the *grand haptic sphere*, we assume that users are able to later perform touch interaction in the same head-coordination manner, which allows them to redeem the visual-haptic alignment.

Several advantages of the acquisition of a *grand haptic sphere* exist. First, the calibration remains a single process. Secondly, the *grand haptic sphere*, being a collection of all of the haptic spheres, provides a much bigger sphere than a single haptic sphere, particularly for the *by-HMD* interface (see Figure 3bd). Third, the usage of the *grand haptic sphere* can work on all three attachment methods without modification, and it can also work for other attachment methods via the user body. The downside, however, is that touch interaction has to undergo a head-coordinated touch interaction, which fortunately is by and large a common and natural behavior during touch interaction. In Study 1, we showed that users can easily adapt their touch behavior with brief practice.

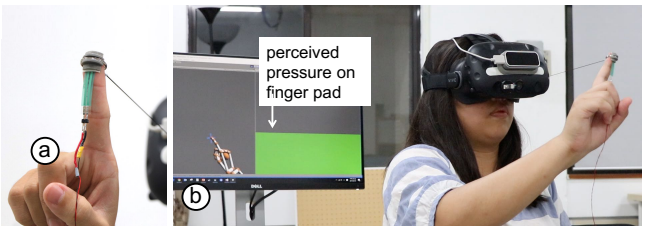

Figure 5: Experiment setup of Study 1. (a) A pressure sensor was placed between the pad and the user finger to record the perceived pressure. (b) Real-time display of pressure and tracking hand to ensure the detection was working well during study.

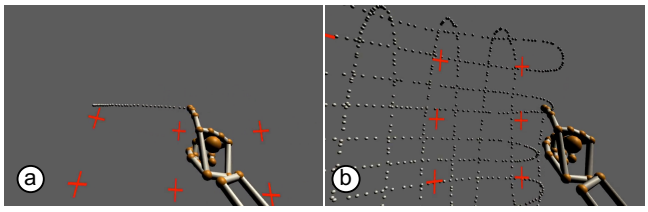

Figure 6: Calibration process. (a) The crosshairs shown in the HMD view indicates the range of the calibration to cover. The user starts from the left-top corner, and (b) draw a continuous stroke horizontally to the bottom and then vertically to provide a good coverage of the calibration region with touch points.)

It is worth noting that the calibration procedure can be understood as taking a snapshot of an individual user's body-touch interaction associated with a string. The procedure captures the user's own way of executing head-coordinated touch interactions. The only requirement for a good alignment in a later touch interaction using the *grand haptic sphere* is that users should adopt a similar overall touch behavior to redeem the alignment. In other word, if a user performs calibration with a fixed head posture, he or she shall redeem the alignment *only* with the same head posture.

5 User Studies

To validate the proposed *grand haptic sphere* and to understand the benefit of physical support for touch interaction in mobile settings, we conducted two user studies.

In Study 1, we tested whether participants could repeat touch behaviors captured in a *grand haptic sphere* acquired during the calibration. In doing so, users' touch interaction can effectively receive physical support on the virtual targets that the *grand haptic sphere* suggests. We measured this by comparing three interface conditions: the *by-HMD*, *by-neck*, and *by-shoulder* conditions, with the *no-support* condition being the baseline condition. For validation, we measured the touch precision and pressure that occurred when virtual targets were pressed, as well as the levels of touch overshooting. The results showed that the calibration effectively captured the surface in all three attachments, considering the touch precision, perceived pressure, and overshooting errors. The *by-HMD* interface showed the best matching in that it had the smallest overshooting error (M: 7.3mm; SD: 2.74mm). The study also informed an effective button size of 22 mm by 22 mm for 95% accuracy. The *by-HMD* interface was selected in Study 2 to explore the benefit of physical support in keeping up the touch precision during mobile situations.

In Study 2, we compared touch accuracy with and without physical support in sitting and walking situations. The results suggested that physical support significantly improves touch precision under walking situations.

5.1 STUDY 1: Feasibility Study and Pointing Accuracy

The goal was to measure the effectiveness of the calibration procedure and effective button sizes given our implementation using LeapMotion.

5.1.1 Apparatus

The same setup as in the implementation section was used. To ensure that the participants perceived the proper physical support from the string while reaching with the finger the virtual targets on the acquired *grand haptic sphere*, we added a pressure sensor on the finger pad to record the pressure profile for each finger touch (Figure 5). The personal computer used in the study was an Intel i7-6700 Processor with a 32GB RAM running Windows 10.

5.1.2 Interface

The study followed a within-subject design with a single factor *interface*, including the *by-HMD*, *by-neck*, and *by-shoulder* conditions, as well as the *no-support* condition as the baseline condition.

For the task we used 12 crosshairs laid out in a 4x3 pattern on the *grand haptic sphere* acquired during calibration. Regardless of the size of the *grand haptic sphere*, columns and rows in the grid were spaced 25 degrees apart; in total, the crosshairs spanned 75 degrees horizontally, and 50 degrees vertically. The crosshairs were at 80 mm squared in size and 5 mm in thickness.

During calibration, a haptic sphere 35 cm in diameter was initially used to lay out the crosshairs to provide visual guidance. The initial sphere was positioned differently in each of the three attachment conditions to fit a preferred interaction space for the user, as illustrated in Figure (2)def. As such, the sphere in the *by-HMD* condition was positioned at the user's eye (e.g., the viewpoint of the virtual camera) so that a target would appear in front of the user at a distance of the string length. The sphere in the *by-neck* condition was positioned 10 cm below and was further rotated 30 degrees toward the side of the attachment on the shoulder in the *by-shoulder* condition. Participants drew a continuous stroke along all crosshairs (Figure 6). Once the calibration was done, the estimated *grand haptic sphere* was used in the study.

We intended to have the targets laid out wide to test the proposed *grand haptic sphere* algorithm; however, in such layout, a target may appear outside of the user's viewport. To avoid the search for targets, we displayed a halo effect [8] for off-viewport targets, which participants could simply follow as a guide for locating the target (Figure 7).

5.1.3 Procedure

Participants were each seated on a chair during the experiment. After the study debriefing, they received help with putting on the interface as required by the condition. For each participant, the *grand haptic sphere* was computed at the beginning of each condition; crosshair targets were then laid out accordingly. In the *no-support* condition, participants first acquired the sphere with the *by-HMD* interface and then proceeded with the task without the string.

Once the calibration was completed, participants tested the system for few trials to familiarize themselves with the touch feedback. Meanwhile, the experimenter ensured the calibration was correct.

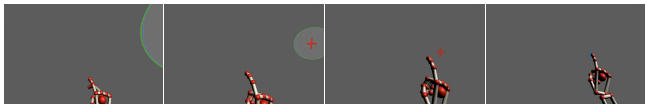

Figure 7: Halo effect to help locate an offscreen target for the tasks of target acquisition.

Before the study trials started, participants were informed to perform head-coordinated touch interaction as they did in the calibration.

For each trial, a crosshair was rendered in red. Users performed a *take-off selection* [7] on crosshair targets as accurately as they could. If a finger entering a crosshair region was detected, the crosshair turned blue. The system advanced to the next trial after each click attempt, regardless of whether it was correct. Therefore, when overshooting a target (e.g., the finger passed through the 5-mm-thick crosshair), the target target was completed on the backside. We recorded the take-off position on the target (ignoring the 5-mm-thickness) as the resulting touch position. The task time was recorded from the start of a trial until a selection was made. We also recorded the pressure profile of each touch and the complete touch trace to measure the size of the overshooting clicks.

Each of the 12 crosshair locations was repeated eight times; the presentation order was randomized. The interface condition order was balanced with the Latin square design. The total click trials included 384 clicks (4 interfaces x 12 crosshairs x 8 repetitions) per participant. The study took approximately 60 minutes.

5.1.4 Participants

Nine participants (three females) from our institute were recruited, aged 22-24 (M:23; SD: 0.67). Two participants were left-handed, and all used touchscreen smartphones. Five of them had experience with virtual reality gaming.

5.2 Results and Discussion

5.2.1 Precision and Pressure in Take-Off Selections

Figure 8a shows the average precision in target acquisition (e.g., distance to crosshair centers in mm) for the instances of take-off selections completed. All conditions showed similar offsets and standard deviations, indicating that users performed target acquisitions equally well without problems. Figure 8b shows the average perceived pressure value measured by the pressure sensor on the finger. To make sense of the pressure values, we also provide the averaged peak pressures extracted from each trials.

All targets were successfully accomplished using take-off selection, indicating that the physical support did not pull-stop the finger above the targets. The pressure chart further provides evidence that a physical support indeed appeared in each take-off selection. Together, we confirmed that the participants completed the study with physical support in all attachment methods, and that the *grand haptic sphere* appropriately captured the user's touch behaviors.

5.2.2 Touch Overshooting

Touch overshooting is a well-known phenomenon in midair touch interaction previously reported in [10]. Figure 8c captures the same phenomenon in our study, with the *no-support* condition having the highest overshooting error at $25.17mm$ ($SD : 9.05mm$).

All haptic conditions alleviated overshooting to some extent, where the *by-HMD* condition ($7.3mm; SD : 2.74mm$) had a lower error rate than the *by-neck* ($12.89mm; SD : 3.77mm$) and the *by-shoulder* ($12.88mm; SD : 6.67mm$) conditions did. This can be partly explained by the degree of the extension of the string allowed in each of the three attachment conditions, which contributed to the overshooting differently. In the *by-HMD* condition, no extension was allowed due to the HMD's rigid body, but a small extension was attributed to the fingertip's skin tissue. In the *by-neck* condition, the extension was found in the neck skin, which created a greater deformation. In the *by-shoulder* condition, the backpack strap's elasticity caused the extension when the string was pulled.

On the other hand, the results of overshooting can also be explained by the force transfer in different attachment methods, which we explain with the pressure profile in the next section.

5.2.3 Pressure Profile in Detail

To further investigate the force perceived at the finger in various attachment conditions, we plot a pressure profile for trials at position #6 (find in Figure 9, right), from six of our participants (Figure 10). Each curve illustrates the pressure profile of a finger approaching a target (from positive values to zero on the x-axis) and finishing with smaller (*by-HMD* condition) or greater (*by-neck* and *by-shoulder conditions*) overshooting distances (e.g., the minus values on the x-axis).

The small force, also with a smaller standard deviation, captured on the finger (refer to Figure 8b) in the *by-HMD* condition was due to a more efficient force transfer allowed by the rigidness of the HMD device that shortened the user's reaction to the overshooting, which contributed to a reduced overshooting distance. The users reported that the perception of force in the *by-HMD* condition was sharp and instantaneous.

In comparison, the greater force in the *by-neck* condition was due to the slower force transfer caused by the soft skin of the neck. This progressive force at the finger was less perceivable initially, inviting the finger to advance further until the stop, which leads to a greater overshooting distance. The *by-shoulder* condition seemed to produce the highest force, because not only did the attachment on the backpack strap's deformation slow the force transfer, but also the strap largely absorbed the force before the user perceived it on the shoulder. In addition, these plots also reveal great interpersonal differences in touch behaviors. For example, P1 had light tapping, whereas P6 pressed harder than the others did.

5.2.4 Overshooting Distribution

Figure 9 displays overshooting distances for each target location. In all haptic conditions, the overshooting errors seem slightly larger in the center compared with those in the outer regions, indicating that the *grand haptic sphere* matches better on the two sides. A possible reason is that the touch points collected were two times more on the sides given our uniform sampling process. The best-fit sphere therefore tended to align better for peripheral targets. To deal with this issue, one can give higher weight to touch points in the central region or model the points using an ellipsoid instead of a sphere for a *grand haptic sphere* to better capture the shape of the touch points.

5.2.5 Effective Button Sizes

The precision reported in Figure 8a indicates the effective button sizes for the implementation using LeapMotion. Note that this precision is not dependent on our calibration method because physical support does not help in aiming targets, as revealed by the comparable precision and completion time across all conditions (Figure 8d). The average precision across all four conditions is 10.4mm (SD: 5.87mm), suggesting an effective button size at 22mm by 22mm for 95% accuracy.

5.2.6 Summary

The results suggest that the proposed *grand haptic sphere* successfully modeled user touch behaviors with head-coordinated touch interaction, and it equipped users with a greater spherical touch surface. Because the *by-HMD* interface showed the best touch performance in terms of precision and overshooting distance, we adopted the *by-HMD* interface and the effective button size in Study 2 to explore how physical support improves touch interaction in a walking condition.

5.3 STUDY 2: Toward Mobility

The goal of this second study was to understand how physical support assists users in target acquisition during sitting and walking conditions.

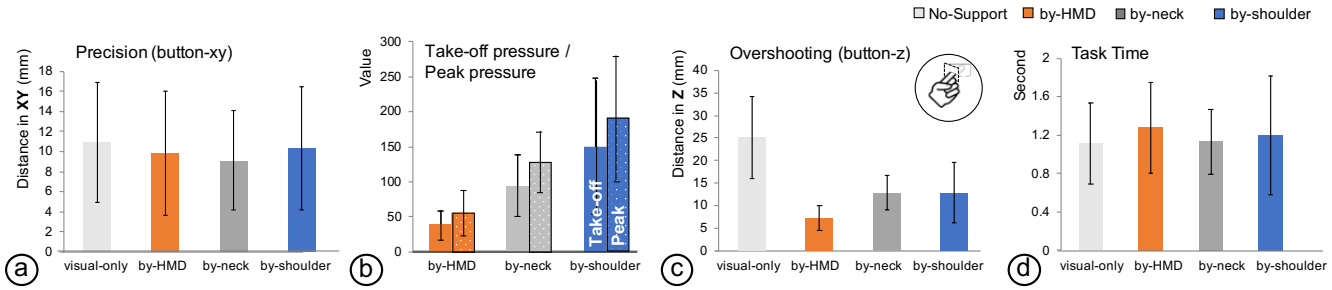

Figure 8: Plot of the precision (mm), perceived pressure on take-off selection (value), and task time (second) with standard deviation.

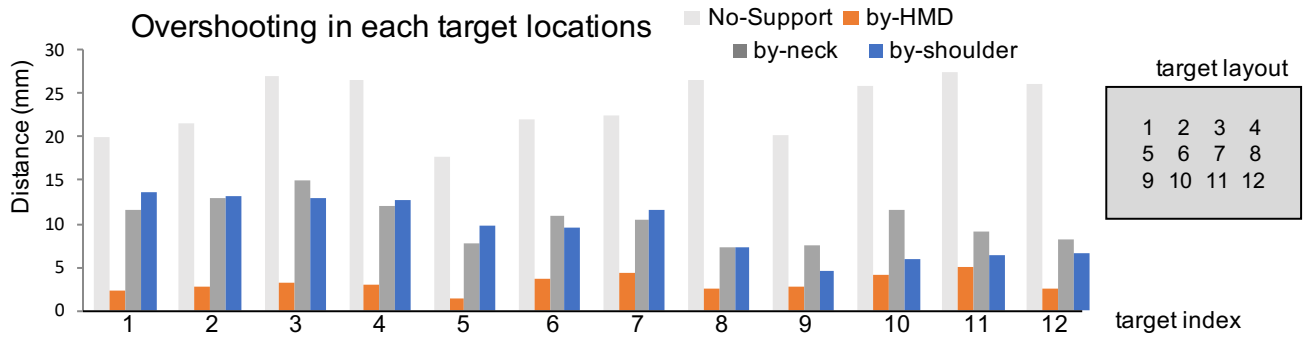

Figure 9: Plot of the overshooting error (mm) in four interface conditions. Corresponding target index in the layout (right).

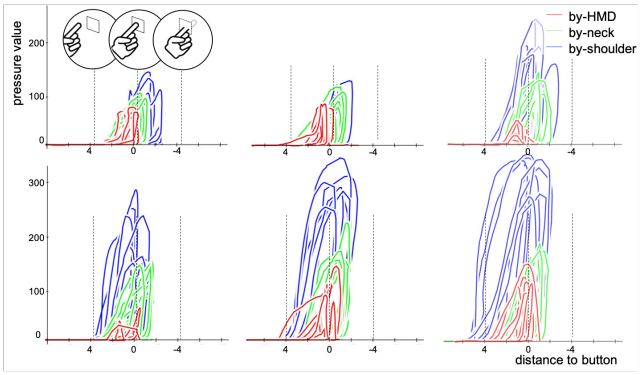

Figure 10: Pressure profiles of six participants. Each curve illustrates the pressure responses before, on, and after the finger arriving at a target at center (e.g., button #6) in three attachment methods.

5.3.1 Interface

This study followed a 2x2x4 within-subject factorial design, with three independent variables: mobility (walking and sitting), haptics (with support and no-support), and button size (25mm, 30mm, 35mm, and 40mm). Physical support was enabled with the *by-HMD* interface.

We used the same 12 target positions laid out in a 4x3 pattern to collect touch accuracy, except that this time, the targets were buttons in varying sizes and not crosshair cursors. For each trial, a button was rendered in white. If a finger entering the button was detected, the button turned blue. Users did *take-off selections* on this button as accurately as they could. If a selection failed, the system beeped to indicate that the trial had failed. The system then advanced to the next trial after each click attempt, regardless of whether the input was successful.

Between trials, we placed a dwell-button in front of the participant for homing. When the participant faced it, the dwell-button counted two seconds and advanced to the next trial. This homing action reset participants to the front after each trial, thus allowing us to control the trial distance. This also helped participants to restore their balance for the next trial. In the walking condition, a Chanson CS-8830 treadmill was used. We set a walking speed at 3.2 km/h as suggested by previous work [23]. The speed was measured at 3.06 km/h.

5.3.2 Apparatus

The same HMD device and the string attachment of the *by-HMD* interface were used, except that a mixed-reality mode instead of a virtual reality mode was presented to the participant.

The mixed-reality mode was enabled by blending three layers of images together as displayed in Figure 11. These layers from the bottom to the top layers are the (1) context layer: the mono color image from HTC Vive's built-in camera, (2) stereo-hand layer: the stereo infrared images from LeapMotion's stereo cameras, and (3) graphics-hand layer: the graphic hand provided by the LeapMotion tracker.

The context layer allowed users to see the real environment, but it appeared flat in the viewport because the same video image was presented to both eyes. If we had added LeapMotion's graphics hand directly on top of the context layer, some visual separation would have occurred between the skin-hand in the context layer and the graphics hand, resulting in double-hand confusion to users. To remove this confusion and give depth perception to the user's

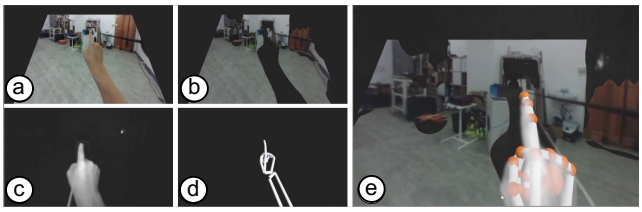

Figure 11: The image process to simulate a mixed-reality view in the HMD by blending (a)(b) a context-layer with skin hand removed using Vive's front camera, (c) a stereo-hand layer, and (d) a graphics-hand layer from LeapMotion. (e) The mixed-reality view present to users.

hand, we added a stereo infrared hand on top of the context layer, achieved by first removing the user's skin hand shown in the context layer and then painting the skin color pixels black (Figure 11b). In the stereo-hand layer, we set each pixel's alpha value using its intensity, which kept only the bright pixels (e.g., user hand) in the image and made transparent the dark pixels mainly pertaining to the environment. In the graphics-hand layer, the graphics hand from LeapMotion was directly used. The image process for producing the context and stereo-hand layers was achieved using a shader in the Unity3D engine. Our implementation of the mixed-reality mode based on was achieved at 90 frames per second. However, the image frame rate was limited by the Vive's camera frame rate at 55 Hz.

5.3.3 Procedure

In both the sitting and walking conditions, the same calibration *procedure* as in Study 1 was used. In the walking conditions, an extended training session was provided. All participants had no prior experience of walking on the treadmill with a mixed-reality HMD. Our pilot study confirmed that most participants can balance themselves within three minutes of practice.

The training in each walking condition consisted of two sections. In the walking-practice section, we had participants focus on walking from a slow speed of 2km/h up to 3.2km/h after 30 seconds. At the target speed, the participant walked for three minutes alone and was asked whether he or she was comfortable with the speed in mixed-reality mode. In the second walking-acquisition section, the participant first did the calibration to acquire a *grand haptic sphere* while standing on the treadmill. Then, we had him or her start to walk on the treadmill at the target speed for 15 seconds. Then, the participant practiced for 12 (4x3) trials, and reported whether he or she was comfortable with continuing the study trials. The practice continued until the participant was confident about the task. Out of all of the participants in this study, only two participants had to repeat the second section once more before they were confident. For safety reasons, the experimenter would place an arm behind the participant's back around the end of the treadmill runway (Figure 12b) to give a haptic cue when the participant was walking too slowly during the trials.

The tasks comprised three blocks. In each block, 48 buttons (4 button sizes x 12 button positions) appeared once; the presentation order was randomized. The mobility conditions order was balanced following the Latin-square design, with the haptics conditions balanced within the mobility condition. Between blocks, the participant took a three-minute break standing on the treadmill. The total number of trials was 576 (2 mobility x 2 haptics x 4 sizes x 12 button positions x 3 blocks) per participant.

The walking condition took 90 minutes and the sitting condition took 40 minutes, including training. Because the walking condition was more exhausting for participants, the two mobility conditions for each participant were conducted at least two hours apart. All participants were compensated with 10 USD in local currency.

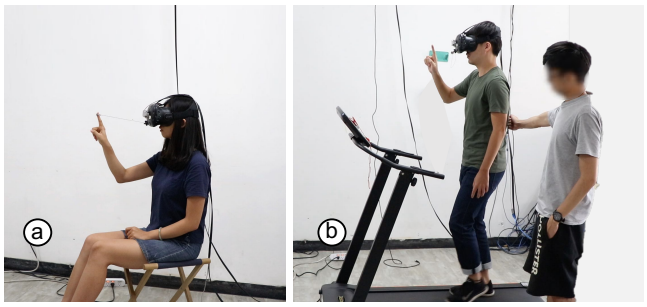

Figure 12: Experiment setup of (a) sitting and (b)walking conditions in Study 2. In walking condition, the experimenter provides a haptic cue with his arm if the participant was walking behind the treadmill speed on tasks. This photo is retouched to remove clutter in the background.

5.3.4 Participants

We recruited 13 participants (2 females), aged 22-36 (M: 24.3; SD: 3.84). All participants were healthy and right handed. None of them had experienced touch interaction with mixed-reality HMDs.

5.4 Results and Discussions

5.4.1 Target Acquisition Accuracy

The data were analyzed using repeated-measures analysis of variance (ANOVA) and Bonferroni-corrected paired t-tests for pairwise comparisons. ANOVA yielded a significant effect of mobility ($F_{1,12}$ = 39.75, $p < .00001$), haptics ($F_{1,12}$ = 21.56, $p < .001$), and button size ($F_{3,36}$ = 25.86, $p < .0001$). An interaction did take place between mobility and haptics.

Figure 13a shows the error rate of mobility and haptics across button sizes. An interaction was found between mobility and haptics ($F_{1,12}$=5.55, $p < .05$), where physical support had a smaller impact when participants were sitting. Error was significant higher when participants were walking (M = 9.7%, SD = 5.6%;) than sitting (M = 4.6%, SD = 2.5%; $p < .0001$), and when they were without physical support (M = 8.6%, SD = 6.0%) than with support (M = 5.7%, SD = 3.3%; $p < .001$). In the two sitting conditions, there was no difference between whether or not physical support was presented (with support: M = 5.1%, SD = 2.8%; w/o support: M = 4.1%, SD = 2.2%), but in the walking conditions, error was significantly lower with support (M = 7.2%, SD = 3.5%) than with no support (M = 12.1%, SD = 6.5%; $p < .005$).

An interaction was also found between mobility and button size ($F_{3,36}$=6.05, $p < .005$). Figure 14a revealed that the mobility had a larger impact when buttons were smaller. We further examined error rates in sitting and walking conditions against button sizes, respectively. A significant difference was found between buttons in the walking condition ($F_{3,36}$ = 18.95, $p < .00001$), but only a borderline difference was found in the sitting condition ($F_{3,36}$ = 4.46, $p < .05$). This indicates that the impact of the mobility on button sizes was mainly relevant in the walking conditions.

Then, we compared the physical support against the button size in the walking condition only, as shown in Figure 14b. No interaction was found between haptics and button size in the walking condition ($F_{3,36}$ = 2.15, $p = .11$). Significant main effects of both haptics ($F_{1,12}$ = 14.05, $p < .005$) and button size ($F_{3,36}$ = 18.86, $p < .0001$) were found, indicating that the physical support outperformed no support (visual-only) under the walking condition.

5.4.2 Task Time

Figure 13b shows the task time mobility and haptics across button sizes. We count the time using only the correct trials. No significant differences were found in mobility, haptics, and button size. In the

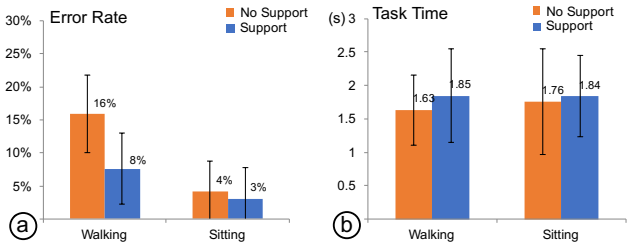

Figure 13: Plot of average error rate and task time of Mobility and Haptics across button sizes. The error bars are standard deviations.

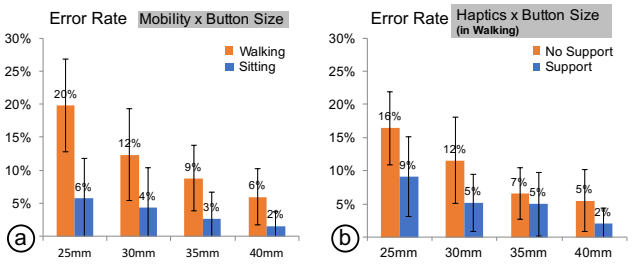

Figure 14: (a) Plot of error rate regarding Mobility and Button Size. (b) Plot of error rate regarding Support and No Support conditions when in walking condition.

walking condition, the *no support* condition seemed slightly faster than the *support* condition was, but it also had significantly more errors made than the *Support* condition (16% vs. 4%) did, indicating that without physical support, participants were unable to correct a touch via take-off selections, which took less time in each task. In fact, they had no chance to alter a touch, as overshooting touches had completed a task by passing through it. In comparison, with physical support, participants could inspect if a touch was successful before taking-off the touch to complete a selection. If a touch did not land at the button initially, they could correct it by moving the touch onto the button, which took longer time but led to fewer errors.

6 Discussion and Limitations

The results showed that physical support stabilizing midair touch effectively improved the touch accuracy particularly in the walking conditions. We also found that this support does not negatively impact the performance in terms of task time. However, the result was confirmed only with the setup of attaching the string on the HMD, which had sharper haptic feedback than when the string was added to the user's neck or shoulder. Further study is required to determine how the stiffness of physical support may affect the user's performance.

Scalability Although we have presented three ways for adding the interface to the HMD, the user's neck, and the users shoulder, they are not interchangeable because the string length needs to be predefined and fixed. Moreover, the proposed simple mechanism of using a passive string limits users to a static spherical touch interface. The above limitations can be dealt with by augmenting the cord reels with a breaking mechanism, similarly to SPIDAR-S [25], allowing for the adjustment of the string length for various wearable configurations and spherical sizes.

Ergonomic Concern Our current prototype adopted a finger pad that grips the user's finger and may obstruct the user from lowering his/her hand during interaction. An alternative design can work around this issue by using the retractable reel body as graspable. When the string is attached to the HMD, for example (Figure 15), users grab the reel as a stylus for interacting with the HapticSphere. Releasing the reel automatically retracts it to the home position on the HMD and disengages the interaction.

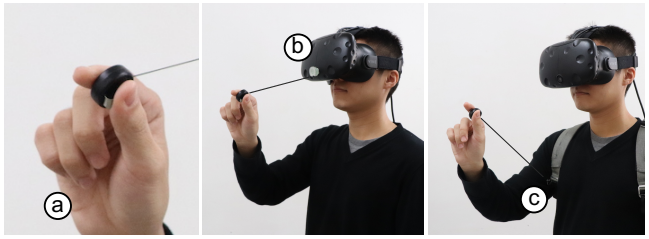

Figure 15: An alternative design using the retractable reel as a graspable stylus to allow fast engage and disengage of spherical interface.

Haptic Property of Physical Support The perceived haptic sphere depends on the material properties of the string used (e.g., flexible vs rigid strings). However, the visual interfaces do not need to perfectly mimic the physical stiffness of the underlying haptic constrain, and by using a retargeting techniques, such as the Sparse Haptic Proxy [12], the visual channel can compensate for inaccuracies in the haptic rendering. Our future work will investigate the user's perception of using manipulated visual information to compensate for distortions in the haptic channel.

Eye-hand Coordination Finally, the acquisition of a *grand haptic sphere* allows users to operate with a large surrounding spherical surface using a *head-coordinated touch interaction*. In our studies, we found that users can easily adapt to this touch behavior after a short practice. However, our studies do not account for the case when eye and hand motions are discoordinated, for instance, when the user is looking to a different target and attempts to touch another target location. Fortunately, this only causes a small displacement to the haptic-visual alignment depending on the targets' distance from the viewport center. In further work, we plan, however, to remove the head-coordinated constrains, using a gaze tracker that tells the user's eye direction and compensates for any displacement.

7 Conclusion

We have presented *HapticSphere*, a wearable string-based passive haptic interface and a single-step calibration method used to acquire a *grand haptic sphere*, which combines provide physical support and feedback for in-air selections during a mobile mixed-reality experience. With two studies we demonstrated that physical support enables precise touch interaction using in-air selections in mobile situations (e.g., walking). In future work, we plan to integrate the *grand haptic sphere* with haptic retargeting techniques for richer 3D interactions. We also plan to integrate gaze trackers to remove the constraint of eye-hand-coordinated touch.

Acknowledgments

This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST106-2628-E-009-007-MY2, 106-2923-E-002-013-MY3, 107-2218-E-002-009, 107-2218-E-011-016). Andrea Bianchi was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) No. 2018R1A5A7025409.

References

[1] Cyberglove systems inc. cybergrasp glove. http://www.cyberglovesystems.com/cybergrasp.

[2] M. Achibet, A. Girard, M. Marchal, and A. Lcuyer. Leveraging passive haptic feedback in virtual environments with the elastic-arm approach. *Presence*, 25(1):17–32, July 2016. doi: 10.1162/PRES_a_00243

[3] M. Achibet, A. Girard, A. Talvas, M. Marchal, and A. Lcuyer. Elastic-arm: Human-scale passive haptic feedback for augmenting interaction and perception in virtual environments. In *2015 IEEE Virtual Reality (VR)*, pp. 63–68, March 2015. doi: 10.1109/VR.2015.7223325

[4] M. Achibet, M. Marchal, F. Argelaguet, and A. Lcuyer. The virtual mitten: A novel interaction paradigm for visuo-haptic manipulation of objects using grip force. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 59–66, March 2014. doi: 10.1109/3DUI.2014.6798843

[5] K. Akahane, J. Hyun, I. Kumazawa, and M. Sato. Two-handed multi-finger string-based haptic interface spidar-8. In I. Galiana and M. Ferre, eds., *Multi-finger Haptic Interaction*, pp. 109–147. Springer London, London, 2013. doi: 10.1007/978-1-4471-5204-0_6

[6] S. C. Alexis Paljic. A passive stringed haptic system for immersive environments. In *EuroHaptics 2004*, pp. 82–87, 2004.

[7] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1923–1932. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518995

[8] P. Baudisch and R. Rosenholtz. Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pp. 481–488. ACM, New York, NY, USA, 2003. doi: 10.1145/642611.642695

[9] H. Benko, C. Holz, M. Sinclair, and E. Ofek. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 717–728. ACM, New York, NY, USA, 2016. doi: 10.1145/2984511.2984526

[10] L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 2625–2634. ACM, New York, NY, USA, 2010. doi: 10.1145/1753326.1753725

[11] H.-Y. Chang, W.-J. Tseng, C.-E. Tsai, H.-Y. Chen, R. L. Peiris, and L. Chan. Facepush: Introducing normal force on face with head-mounted displays. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pp. 927–935. ACM, New York, NY, USA, 2018. doi: 10.1145/3242587.3242588

[12] L.-P. Cheng, E. Ofek, C. Holz, H. Benko, and A. D. Wilson. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 3718–3728. ACM, New York, NY, USA, 2017. doi: 10.1145/3025453.3025753

[13] I. Choi and S. Follmer. Wolverine: A wearable haptic interface for grasping in vr. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16 Adjunct, pp. 117–119. ACM, New York, NY, USA, 2016. doi: 10.1145/2984751.2985725

[14] A. Delazio, K. Nakagaki, R. L. Klatzky, S. E. Hudson, J. F. Lehman, and A. P. Sample. Force jacket: Pneumatically-actuated jacket for embodied haptic experiences. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 320:1–320:12. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173894

[15] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. doi: 10.1145/358669.358692

[16] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson. 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*, CHI '16, pp. 1991–1995. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858487

[17] J. Gugenheimer, D. Wolf, E. R. Eiriksson, P. Maes, and E. Rukzio. Gyrovr: Simulating inertia in virtual reality using head worn flywheels. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 227–232. ACM, New York, NY, USA, 2016. doi: 10.1145/2984511.2984535

[18] S. Heo, C. Chung, G. Lee, and D. Wigdor. Thor's hammer: An ungrounded force feedback device utilizing propeller-induced propulsive force. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pp. D110:1–D110:4. ACM, New York, NY, USA, 2018. doi: 10.1145/3170427.3186544

[19] R. Hinchet, V. Vechev, H. Shea, and O. Hilliges. Dextres: Wearable haptic feedback for grasping in vr via a thin form-factor electrostatic brake. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pp. 901–912. ACM, New York, NY, USA, 2018. doi: 10.1145/3242587.3242657

[20] M. Hirose, K. Hirota, T. Ogi, H. Yano, N. Kakehi, M. Saito, and M. Nakashige. Hapticgear: The development of a wearable force display system for immersive projection displays. In *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, VR '01, pp. 123–. IEEE Computer Society, Washington, DC, USA, 2001.

[21] S. Je, H. Lee, M. J. Kim, and A. Bianchi. Wind-blaster: A wearable propeller-based prototype that provides ungrounded force-feedback. In *ACM SIGGRAPH 2018 Emerging Technologies*, SIGGRAPH '18, pp. 23:1–23:2. ACM, New York, NY, USA, 2018. doi: 10.1145/3214907.3214915

[22] H. Kim, M. Kim, and W. Lee. Hapthimble: A wearable haptic device towards usable virtual touch screen. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 3694–3705. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858196

[23] R. Knoblauch, M. Pietrucha, and M. Nitzburg. Field studies of pedestrian walking speed and start-up time. *Transportation Research Record: Journal of the Transportation Research Board*, 1538:27–38, 1996. doi: 10.3141/1538-04

[24] P. Lopes, A. Ion, and P. Baudisch. Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pp. 11–19. ACM, New York, NY, USA, 2015. doi: 10.1145/2807442.2807443

[25] S. Ma, K. Akahane, and M. Sato. String-based force display for mobile haptics. In *SIGGRAPH Asia 2015 Haptic Media And Contents Design*, SA '15, pp. 15:1–15:2. ACM, New York, NY, USA, 2015. doi: 10.1145/2818384.2818402

[26] K. Minamizawa, S. Fukamachi, H. Kajimoto, N. Kawakami, and S. Tachi. Gravity grabber: Wearable haptic display to present virtual mass sensation. In *ACM SIGGRAPH 2007 Emerging Technologies*, SIGGRAPH '07. ACM, New York, NY, USA, 2007. doi: 10.1145/1278280.1278289

[27] J. Rekimoto. Traxion: A tactile interaction device with virtual force sensation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pp. 427–432. ACM, New York, NY, USA, 2013. doi: 10.1145/2501988.2502044

[28] J. Shigeyama, T. Hashimoto, S. Yoshida, T. Aoki, T. Narumi, T. Tanikawa, and M. Hirose. Transcalibur: Weight moving vr controller for dynamic rendering of 2d shape using haptic shape illusion. In *ACM SIGGRAPH 2018 Emerging Technologies*, SIGGRAPH '18, pp. 19:1–19:2. ACM, New York, NY, USA, 2018. doi: 10.1145/3214907.3214923

[29] E. Strasnick, C. Holz, E. Ofek, M. Sinclair, and H. Benko. Haptic links: Bimanual haptics for virtual reality using variable stiffness actuation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 644:1–644:12. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3174218

[30] C. Swindells, A. Unden, and T. Sang. Torquebar: An ungrounded haptic feedback device. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pp. 52–59. ACM, New York, NY, USA, 2003. doi: 10.1145/958432.958445

[31] S.-Y. Teng, T.-S. Kuo, C. Wang, C.-h. Chiang, D.-Y. Huang, L. Chan, and B.-Y. Chen. Pupop: Pop-up prop on palm for virtual reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pp. 5–17. ACM, New York, NY, USA, 2018. doi: 10.1145/3242587.3242628

[32] A. Zenner and A. Krger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1285–1294, April 2017. doi: 10.1109/TVCG.2017.2656978