Kine-Appendage: Enhancing Freehand VR Interaction Through Transformations of Virtual Appendages

Yang Tian, Hualong Bai, Shengdong Zhao, Chi-Wing Fu, Member, IEEE, Chun Yu, Senior Member, IEEE, Haozhao Qin, Qiong Wang, and Pheng-Ann Heng, Senior Member, IEEE

Abstract—Kinesthetic feedback, the feeling of restriction or resistance when hands contact objects, is essential for natural freehand interaction in VR. However, inducing kinesthetic feedback using mechanical hardware can be cumbersome and hard to control in commodity VR systems. We propose the kine-appendage concept to compensate for the loss of kinesthetic feedback in virtual environments, i.e., a virtual appendage is added to the user’s avatar hand; when the appendage contacts a virtual object, it exhibits transformations (rotation and deformation); when it disengages from the contact, it recovers its original appearance. A proof-of-concept kine-appendage technique, BrittleStylus, was designed to enhance isomorphic typing. Our empirical evaluations demonstrated that (i) BrittleStylus significantly reduced the uncorrected error rate of naive isomorphic typing from 6.53% to 1.92% without compromising the typing speed; (ii) BrittleStylus could induce the sense of kinesthetic feedback, the degree of which was parity with that induced by pseudo-haptic (+ visual cue) methods; and (iii) participants preferred BrittleStylus over pseudo-haptic (+ visual cue) methods because of not only good performance but also fluent hand movements.

Index Terms—Visual kinesthetic feedback, virtual appendage, visual transformation, isomorphic typing

1 INTRODUCTION

Empowering commodity VR systems with hand tracking capability has become a trend, making freehand VR interaction, i.e., interacting with virtual objects using bare hands without extra hardware, available for common consumers. Freehand VR interaction is a promising interaction paradigm due to its convenient and direct characteristics. However, existing freehand VR interaction modalities supported by commodity VR systems are still not sufficiently natural because of the absence of haptic feedback.

Haptic feedback includes tactile feedback (the feeling of touch) and kinesthetic feedback (the feeling of restriction or resistance) [1]. In VR, both rely mainly on extra hardware to generate. Tactile feedback is usually induced by the vibration of physical controllers, whereas kinesthetic feedback is mainly induced by mechanical exoskeleton [2], [3] and electrical muscle stimulation devices [4], [5]. These devices are cumbersome or expensive to be integrated into commodity VR systems. Also, they are hard to control and maintain.

Some researchers applied the pseudo-haptic concept [6] to compensate for the absence of kinesthetic feedback in VR without requiring extra hardware, i.e., as a user’s avatar hand contacts a virtual object, although the corresponding real hand has sunk into the virtual object, the avatar hand is visually constrained by the surface of the virtual object. The pseudo-haptic concept could induce the illusion of kinesthetic feedback [1]. However, this concept can make freehand interaction performance worse due to the obvious difference between the position and pose of the avatar hand and those of the real counterpart, e.g., applying the pseudo-haptic concept is found to significantly increase the error.
distance between the target position and the final position of a virtual object released by an avatar hand [7], [8].

Inspired by the concept of adding a virtual arm to a user’s avatar to guide her freehand interactions [9], we propose a novel concept for inducing the illusion of kinesthetic feedback, kine-appendage. Kine-appendage adds a virtual appendage to the user’s avatar hand. As the appendage contacts a virtual object, the appendage exhibits visual transformations (such as rotation and deformation) similar to those of its counterpart in the physical world under the resistance of the virtual object and the force applied by the avatar hand. After the user withdraws her hand to a particular extent, the appendage recovers its original appearance. See Fig. 1 for some illustrative examples.

To initiate the investigation of the kine-appendage concept, which is defined for general interactions, we designed and evaluated a particular instance for this concept, i.e., the BrittleStylus technique for isomorphic typing. Isomorphic typing refers to using a user’s avatar’s virtual fingers to directly type on a virtual keyboard in mid-air. This basic modality of text entry is user-friendly to all users, including children and the elderly. See details of the BrittleStylus technique in Section 6.

We conducted three user studies to evaluate BrittleStylus. The first one focused on evaluating its performance. The second one investigated whether the color of a virtual tool (BrittleStylus/controller’s ray) affects isomorphic typing performance. The third one focused on investigating the degree to which the illusion of kinesthetic feedback is induced by BrittleStylus. The main contributions in this work are as follows.

- The introduction of the kine-appendage concept and a deeper investigation into an instance of such a concept in isomorphic typing called BrittleStylus. The BrittleStylus technique helped reduce the uncorrected error rate of naive isomorphic typing significantly from 6.53% to 1.92% without compromising the typing speed;
- The degree of the illusion of kinesthetic feedback induced by BrittleStylus was parity with those induced by pseudo-haptic (+ visual cue) methods;
- Participants were very sensitive to the fluency of hand movements in isomorphic typing. They preferred BrittleStylus over pseudo-haptic (+ visual cue) methods because of not only good performance but also fluent hand movements.

2 RELATED WORK

This section discusses the areas of kinesthetic feedback and freehand text entry in VR that are related to our work.

2.1 Adding Kinesthetic Feedback in VR

To address the absence of kinesthetic feedback in VR, researchers have explored various solutions including providing physical proxies for virtual objects, using hardware that generates kinesthetic feedback, and the pseudo-haptic concept.

One body of work [10], [11], [12], [13] utilized physical objects as passive physical proxies to provide kinesthetic feedback for similarly-shaped virtual objects. Physical proxies could also become dynamic through attaching them to robot arms [14], [15], [16] or being held by people [17]. However, this approach is less effective for virtual objects with shapes different from the available physical objects. Besides, a physical proxy could match multiple virtual objects by warping users’ hand movements or the virtual world [18], [19]. However, the extent of warping needs to be limited, or users will notice the warping, which breaks the immersive experience or might induce simulator sickness.

Another body of work [2], [3], [20], [21], [22], [23] aimed to design mechanical exoskeleton devices to provide kinesthetic feedback in VR. Electrical muscle stimulation devices were also adopted to make users feel kinesthetic feedback by actuating the user’s muscles [4], [5], [24], [25], [26], [27]. Besides, mitten gloves could provide kinesthetic feedback via water flowing in an embedded liquid bladder [28] or the layer jamming technique [29]. However, the above methods relied on complex, cumbersome, or relatively high-cost hardware, making them less accessible to the public.

Another approach is to create an illusion of kinesthetic feedback using pseudo haptics [6] without relying on physical objects or any extra hardware. Rietzler et al. [1] applied...
the pseudo-haptic concept to compensate for the absence of kinesthetic feedback, and provided tactile feedback via vibration of physical controllers in a commodity VR system. Later, Rietzler et al. [30] adjusted the offset between the real hand and the corresponding avatar hand according to the extent of bicep muscle tension measured by a Thalmic Myo armband. They also proposed displaying the weight of a virtual object lifted by a user’s avatar hand by deliberately creating a positional offset between the avatar hand and the corresponding real hand [31]. Ban et al. [32] proposed hit-stop visual feedback to provide visual kinesthetic feedback for impact sensations, i.e., pausing movement or displaying a slow-motion animation at the moment of impact. The illusion of perceiving the physical constraint of water could be induced by making the movements of a user’s avatar limbs or head slower than their real ones [33], [34]. Mass and mass distribution of a virtual object with a physical proxy could also be displayed by manipulating the virtual object’s rotational movement or pivot point of rotation when the user rotated the physical proxy [35]. Although the pseudo-haptic concept could compensate for the absence of kinesthetic feedback to some extent without requiring extra hardware, the pseudo-haptic concept can make freehand interaction performance worse [7], [8]. The reason should be that the real and avatar hands are mismatched. Prachyabrued et al. [7] proposed adding visual cues to reveal how much the real fingers penetrate a virtual object to improve the pseudo-haptic concept. However, the visual cues still cannot make the real and avatar hands matched. In contrast to the pseudo-haptic concept that deliberately makes a virtual object constrain the avatar hand visually, the kine-appendage concept makes the virtual object constrain the virtual appendage rather than the avatar hand visually. Hence, the real and avatar hands are matched under the kine-appendage concept, which could be beneficial for enhancing freehand interaction performance.

2.2 Freehand Text Entry in VR

Plenty of works [36] focused on text entry in VR. Text entry in VR could be achieved through various tools or means, such as a tracked pen in a CAVE environment [37], a physical keyboard incorporated into VR [38], a circular touchpad of an HTC VIVE controller [39], a combination of speech and a virtual keyboard [40], etc. Nevertheless, we do not intend to provide a comprehensive review of text entry in VR in this section. Instead, we discuss the area of freehand text entry where our work falls in.

Mainstream commodity VR systems have recently started to support 3D hand tracking via front-facing cameras. The Oculus Quest VR headset has four front-facing cameras to track users’ hands to support freehand interactions [41] with virtual objects. The HTC VIVE Pro VR headset [42] also has two front-facing cameras, and it has an experimental 3D hand tracking SDK [43] released. Besides, a low-cost hand tracking device, Leap Motion [44], is frequently attached to commodity VR headsets to support hand tracking. With this trend, freehand text entry would become possible on commodity VR systems.

Research on freehand text entry based on commodity (or low-cost) hand tracking started early. In 2011, AirStroke [45] enabled users to input text using a thumb to draw the graffiti shapes corresponding to letters in mid-air. To support thumb tracking, AirStroke adopted a webcam to track a user’s thumb painted green by color filtering and blob detection. It required users to learn the graffiti alphabet and was speed-limited. ATK [46] adopted a probabilistic decoder to infer what words a user intended to type by analyzing her finger motion data captured by a Leap Motion device when she typed on an imaginary keyboard in mid-air using ten fingers. Based on the association between the ten fingers and the corresponding letter groups when typing on a QWERTY keyboard without vision, PinchType [47] enabled a user wearing an Oculus Quest to type a word by pinching tips of thumbs with those of other fingers to select letter groups containing the word. Also, a language model was employed to provide the most probable word options for the user to choose. Though experienced typists familiar with ten-finger typing without vision may prefer ATK and PinchType, they may not be user-friendly for inexperienced typists like the elderly or children. Besides, ATK and PinchType depend on probabilistic decoders to infer the words that users intend to type. Hence, they are only applicable to input dictionary words, i.e., words within the decoders’ vocabularies. Hence, they can not deal with non-dictionary text, e.g., usernames, passwords, special words, abbreviations, etc.

Isomorphic typing with one or two fingers is an essential paradigm, as it is familiar to most users after touchscreen devices such as cellphones and tablets become prevalent. Also, it applies not only to dictionary words but also to non-dictionary text. However, Speicher et al. [48] found that the performance of isomorphic typing (speed: 9.77 WPM; error rate: 7.57%) with two index fingers was poor under the hardware setup including a Leap Motion and an HTC VIVE headset. They argued that the main reason for the poor performance was the technical challenges of handling hand self-occlusion, tracking accuracy, etc., with the Leap Motion device.

Besides, some works that relied on high-quality motion tracking systems, e.g., OptiTrack [49], supported isomorphic typing with ten fingers in VR [50], [51], [52]. Dudley et al. [50] found that aligning the virtual keyboard to a physical surface led to significant typing speed gain, suggesting that the absence of kinesthetic feedback in VR was a main reason for the poor performance of isomorphic typing. Also, they found that the error rate under the “ten fingers in mid-air” condition (~ 7.5%) was significantly worse than that under “two index fingers in mid-air” condition (~ 4%), implying that using ten fingers to type without haptic feedback brings too much mental demand. Note that these error rates were still rather high given a high-end hand tracking system (OptiTrack) and a decoder for auto-correction were adopted. Different from that Dudley et al. detected button selection via colliders, Richardson et al. [51] enabled users to type on a physical surface with ten fingers by decoding hand motion data captured by an OptiTrack system into text. Gupta et al. [52] enhanced the performance of ten-finger isomorphic typing by providing tactile feedback on finger-bases or wrists with attached vibrotactile actuators. However, there is a big accuracy gap between the current commodity hand tracking on VR headsets and the high-quality hand tracking provided by OptiTrack [53]. Also,
high-end motion tracking systems are too expensive to be adopted in the commodity VR systems. Furthermore, the above methods that adopted probabilistic decoders [50], [51] could not deal with non-dictionary text, either.

In a word, previous works on isomorphic typing focused on evaluating the naive isomorphic typing under different system setups [48], [50] or providing real kinesthetic feedback via physical surfaces [50], [51] or tactile feedback via vibrotactile actuators [52]. To our best knowledge, this work is the first to enhance isomorphic typing under commodity hand tracking by enhancing users’ fine control ability for finger movement via a virtual appendage with transformations that compensates for the absence of kinesthetic feedback.

3 Kine-Appendage Concept

In Fig. 1, we present several examples of kine-appendage transformations of a virtual stylus appended to the tip of an avatar finger when the stylus contacts a virtual object, depending on the way we append the stylus to the fingertip and the toughness of the stylus.

Hinge or Fix? If the stylus is hinged at the fingertip, it rotates around the joint under the visual physical constraint of the virtual object and the avatar finger (see Fig. 1b); if the stylus is fixed at the fingertip, the stylus deforms (see Figs. 1c, 1d, and 1e).

Brittle or Tough? When a stylus deforms, the visual effect of the transformation is subject to its roughness. If it is extremely brittle, it shatters into invisible powder (see Fig. 1c); if it is not, it bends into a curved one (see Fig. 1d) and breaks when it can not bear the force exerted by the finger (see Fig. 1e).

The potential benefits of the kine-appendage concept are as follows.

- Kine-appendage could induce a sense of kinesthetic feedback. Tools become extensions of our bodies after sufficient proficiency with the tools has been achieved [54]. Hence, after using the virtual appendage proficiently, users could feel that a part of their body perceives kinesthetic feedback when they see the virtual appendage exhibits transformations under the visual physical constraint of a virtual object and an avatar hand.

- The real and avatar hands are matched under the kine-appendage concept. Under the pseudo-haptic concept, when the avatar hand contacts a virtual object, an obvious mismatch between the real and avatar hands is deliberately created by making the virtual object constrain the avatar hand visually. In contrast, under our kine-appendage concept, when the virtual appendage attached to the avatar hand contacts a virtual object, only the virtual appendage is constrained by the virtual object visually, while the avatar hand is not. Besides, the systematic errors of hand tracking of VR systems always introduce certain degrees of mismatch between the real and avatar hands. Nevertheless, such mismatches are nearly imperceptible for users immersed in VR environments created by modern commodity VR systems (e.g., Oculus Quest 2 adopted in this work). Thus, perceptually, users immersed in VR feel that the real and avatar hands are matched using a kine-appendage.

- Kine-appendage could enhance the immersive experience. The unrealistic visual artifact that a hand penetrates objects in VR was found breaking the immersive experience [1], [7]. By applying the kine-appendage concept, the existence of the appendage between an avatar hand and an virtual object provides a buffer zone for reducing the probability of hand-object penetration, increasing the realism of the interaction.

4 Interaction Problems of Isomorphic Typing

Dudley et al. [50] categorized the mistypes of isomorphic typing into three folds: (i) substitution, i.e., an incorrect key is pressed; (ii) insertion, i.e., an additional undesired key is pressed; and (iii) omission, i.e., the desired key is not pressed. We introduce the reasons for the above problems.

We suggest that the absence of kinesthetic feedback was an important reason for insertion and omission. When the user types on a soft keyboard of a physical touchscreen device, the kinesthetic feedback provides direct and clear spatial information of the physical touchscreen relative to the user, e.g., the touchscreen’s position, orientation, etc. Such spatial information aids the user to lift her finger from the touchscreen for a safe distance before moving this finger towards the next target on the touchscreen. In contrast, when the user types on a virtual keyboard in mid-air, the virtual finger easily penetrates the virtual buttons for a substantial distance due to the absence of kinesthetic feedback. For instance, the mean penetration depth of the users’ index fingers were found to be 10 mm in Dudley et al.’s experiment [50].

De-selecting a button is more complex on a virtual VR keyboard than on a touchscreen device’s soft keyboard. If the virtual keyboard is vertical, the user has to first withdraw her virtual finger, such that the front side of her virtual finger can move out of the space behind the virtual button. Further, the user has to continue withdrawing her finger slightly, so that the front side of her virtual finger is not too close to the virtual keyboard for a certain safe distance; otherwise, when the user moves this finger towards the next target button, slight oscillation of her hand may likely result in the insertion problem. We observed that users sometimes could not de-select buttons well, if the finger overly penetrated a button, i.e., although the front side of the finger has been moved beyond the virtual keyboard plane, it was still too close to the plane, especially when the user typed quickly. Besides, Dudley et al. [50] observed that when their participants double-typed characters such as T, L, and O, they often failed to move their fingers out of the space behind the virtual button after the first tap, making the second tap not detected by the system (omission problem).

We suggest that the main reason for the substitution problem was that the accuracy of hand tracking provided by the commodity VR headsets was limited, making the participants unable to precisely manipulate the avatar.
finger to type under the standard 19 mm key spacing setting. A similar finding was reported by Speicher et al. [48].

5 PILOT STUDY

Based on the discussion presented in Section 4, we started to design a kine-appendage technique for improving the naive isomorphic typing. In this regard, we conducted a pilot study to explore which visual transformation(s) introduced in Section 1 (see Fig. 1) are suitable for the isomorphic typing scenario.

Participants. We recruited five participants from campus. Among them, three were male and two were female, aged between 19-22 (mean: 21.1). They are all right-handed and three had VR experience.

Apparatus. We adopted an Oculus Quest 2 VR headset with four front-facing cameras for tracking user’s hands in real-time. We adopted the Unity game engine to develop our experiment software. Due to Oculus Quest 2’s hand tracking ability, Oculus Integration SDK [55] for Unity provided a pair of avatar hands that correspond to users’ own real hands in VR (see Fig. 2). We rendered a vertical virtual keyboard 30 cm away from the front side of the VR headset. The virtual keyboard had a standard size as a real keyboard, i.e., the key spacing was 19 mm. Although a larger virtual keyboard requires a less precise hand tracking technique, it brings more head movements, consequently leading to higher motion sickness and workload [48].

We rendered a thin cylinder (diameter: 2 mm, length: 15 mm) as the stylus. The cylinder consisted of 20 sub-cylinders, so that it could deform. We put a spherical collider for detecting whether the stylus tip contacts a virtual button at the end of the cylinder. The diameter of the collider was the same as that of the cylinder. Besides, we put a small sphere (diameter: 0.01 mm) that could not trigger buttons at the index fingertip of the user’s dominant avatar hand. Given this small sphere, our system could acquire its location, which was also the fingertip location, in real-time. For “hinge,” the direction of the centerline of the stylus was determined by the contact point on the keyboard and the fingertip. For “bend,” the centerline was determined by a Bezier curve that connects the contact point and the fingertip. For “break,” the centerline was determined by the above two control points and a point that joins the two broken segments. When the stylus was “approaching,” the keyboard or recovered its original appearance after the distance between the fingertip and the keyboard reached a threshold distance (20 mm in this pilot study), the direction of the stylus’s centerline was the same as that of the index finger.

Task. For each transformation, a participant’s task was to use a stylus attached to the tip of the index finger of the participant’s dominant avatar hand to tap on the virtual keyboard plane freely for a three-minute session. Once a button was tapped, the stylus would exhibit one of the four visual transformations.

Results. All participants reported that they preferred “shatter” the most. They reported that they could not see the other three transformations (“hinge”, “bend”, and “break”) well because their view direction was nearly parallel to the plane determined by the stylus’s centerline that rotates, bends, or breaks; see the visual effects of “bend” in Figs. 2a and 2b from a side view and the egocentric view. This problem cannot be solved because the visual effects of “hinge”, “bend”, and “break” are always discounted unless the keyboard plane is set to be parallel to the user’s view direction. However, this setting makes the user cannot see the characters on buttons. In contrast, as long as the index finger does not occlude the stylus, participants could see the visual effect of “shatter” well from their egocentric view; see Figs. 3a, 3b, 3c, and 3d for an example. Although the other visual transformations are not that suitable for isomorphic typing, they could be suitable for other scenarios. We will discuss their possible usage scenarios in Section 10.

6 BRITTLESTYLUS

We denote the virtual stylus with the “approach-shatter-recover” mechanism as BrittleStylus.

The procedure of using the BrittleStylus technique (see Fig. 3) to type on a virtual keyboard is as follows.

1) “Move:” moving the avatar index finger towards the target button;
2) “Select:” instead of using the avatar index finger to tap virtual buttons, the user needs to use the tip of the appended stylus to tap buttons; upon a successful tap, the stylus shatters immediately;
3) “De-select:” moving the finger away from the keyboard until the stylus recovers its appearance; the stylus will recover its appearance only if the user moves her finger sufficiently backward from the virtual keyboard above a threshold distance; and
4) repeating the above steps until the text entry is done.

Besides inheriting the common potential benefits of the kine-appendage concept discussed in Section 3, the BrittleStylus technique might bring the following specific benefits for isomorphic typing.

- As using a stylus is more accurate than using a finger in fine control for selecting 2D targets [56], [57], using the appended virtual stylus instead of the avatar index finger to hit the virtual buttons could compensate for the accuracy limitation of the hand tracking of commodity VR headsets, aiming to alleviate the substitution problem.

1. the distance between the centroids of two adjacent buttons on the keyboard

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The “approach-shatter-recover” mechanism aims to mitigate the insertion problem: the “shatter” visual cue tries to give users an illusion of tapping a physical touchscreen with the stylus, and the “recover” visual cue implicitly encourages the user to move her finger backward from the keyboard for a suitable safe distance when de-selecting a button. Also, this design helps indicate the “readiness” of selecting the next button.

In this user study, we investigated whether the BrittleStylus technique could enhance isomorphic typing significantly. We compared the BrittleStylus technique (see Fig. 3) with the naive isomorphic typing technique (see Fig. 4a) and the controller pointing technique (see Fig. 4b), i.e., using a ray cast from the VR controller to aim and select buttons on a virtual keyboard. Also, we focused on the single-finger setting in this user study. We will adapt and evaluate the BrittleStylus technique for the two-finger setting in the future.

Participants. We recruited 18 participants (13 males and five females) aged between 17-21 (mean: 18.2) from the campus. All of them were right-handed. Each participant signed an informed consent form before the user study.

Apparatus. The hardware setup and the virtual experimental environment were the same as those in Section 5. After the BrittleStylus shatters, we imagine that there is an invisible intact BrittleStylus still attached to the tip of the avatar’s index finger. The system calculates the position of the tip of the invisible BrittleStylus in real-time according to the direction of the index finger and the length of the BrittleStylus. Then, the system calculates the distance between the tip of the invisible BrittleStylus and the keyboard plane and compare it with the threshold distance to determine whether the stylus should recover. We developed the two baseline techniques as follows.

Isomorphic typing. To detect if the index finger contacts a virtual button, we put cubic colliders behind all virtual buttons and a spherical collider internally tangent to the tip of the index finger of the dominant avatar hand.

Controller pointing. We adopted the UIHelpers prefab provided by the Oculus Integration SDK to render a ray cast from the Oculus controller. The prefab could detect which virtual button the ray points at. Users pressed the Oculus controller’s trigger to confirm the selection of a button.

If the participant selected a button successfully under all conditions, the system would play a click sound as a kind of acoustic feedback and make the white button grey as a kind of visual feedback. Both kinds of feedback lasted 200 ms.

Determining Parameters. We recruited six participants who would not participate in the formal user study from campus to try the BrittleStylus technique and determine the stylus length, stylus diameter, and threshold distance. We rendered a menu below the virtual keyboard. The menu consisted of six buttons, each two of which were used for increasing or decreasing one of the three parameters. The task for each participant was to use the BrittleStylus attached to the index fingertip of their dominant avatar hand to type on a virtual keyboard freely for ten minutes. At the same time, they could use the index finger of their non-dominant avatar hand to tap the buttons in the menu to adjust the parameters. They were told to adjust the parameters to balance the typing accuracy.
and finger-keyboard penetration and make the stylus easy to be noticed. The adjustment granularity for both stylus length and threshold distance was 1 mm, while that for the stylus diameter was 0.5 mm. The initial values of the three parameters were 15 mm, 2 mm, and 10 mm, respectively, which were determined by the experimenter according to his preference. The ranges (minimums to maximums) of the participants’ preferred stylus length, diameter, and threshold distance were (7 mm, 26 mm), (1.5 mm, 3.5 mm), and (8 mm, 16 mm), respectively. The means were 19.3 mm, 2.6 mm, and 12 mm, respectively. We adopted these means in our formal evaluation. For isomorphic typing, the diameter of the spherical collider was 10 mm, which was the mean thickness of fingertips of the above six participants. We determined the diameter by referring to Speicher et al.’s and Dudley et al.’s experiment setups, making the experience of isomorphic typing similar to that of using real hands to type on a physical touchscreen. For controller pointing, the diameter of the controller’s ray was the same as that of the BrittleStylus.

Task. Each participant was asked to transcribe short sentences using different techniques as accurately and as fast as possible. The procedure of the experiment for each technique is described as follows: (i) the participant first practiced transcribing an example sentence displayed above the virtual keyboard to get familiar with the technique; (ii) after the practice, the system displayed a textual hint to ask the participant to rest for 15 seconds to alleviate the gorilla arm syndrome; (iii) after the rest, the “start” button appeared between the participant and the keyboard; once the participant tapped the “start” button, it disappeared, and the formal experiment started; (iv) the participant transcribed the sentence shown above the virtual keyboard; (v) the participant tapped the “next” button to move to the next sentence; since we did not provide a “backspace” button, participants could not correct their mistypes; (vi) the system displayed a textual hint to ask the participant to rest for 15 seconds; and (vii) the participant repeated steps (iv) to (vi) until the whole task was completed.

Experimental Design. We compared the BrittleStylus technique (BS) with two baseline conditions: isomorphic typing (IT) and controller pointing (CP). Each participant performed the task described previously under the three conditions. We collected the following metrics for evaluating the techniques: (i) uncorrected error rate, which was calculated based on Soukoreff et al.’s method; (ii) speed, which was measured in words per minute (WPM); WPM was calculated by dividing the number of transcribed words (any 5-character string) by the time it takes to transcribe them; (iii) the time it takes to transcribe text was from the moment a participant tapped the “start button” to the moment she tapped the “next button” after she finished typing the last sentence, excluding the time for rest; (iii) error numbers of substitution, insertion, and omission, which were calculated based on Levenshtein distance; (iv) penetration depth, which was how much a participant’s virtual finger pad penetrated a virtual button when she tapped it; (v) penetration number, which was the number of finger-keyboard penetration; and (vi) workload based on NASA-TLX.

We conducted the user study for three consecutive days. Each day, participants were asked to transcribe a set of fifteen short sentences (each consisted of 20-25 characters). We prepared three sentence sets for the three days, respectively. The total lengths of the three sentence sets were 374, 372, and 378, respectively. Each experiment day involved all three techniques, the order of which was counterbalanced using a balanced Latin square. All techniques shared the same three sets of sentences mentioned above. After the experiment in the final day, the experimenter talked with each participant and recorded their comments on their experience during the user study.

Uncorrected error rate. Significant difference between the three techniques was found ($\chi^2(2) = 28.778, p < 0.001$). The uncorrected error rate under IT (M = 6.53%, SD = 3.2%) was significantly higher than that under CP (M = 1.55%, SD = 0.98%, Z = -3.724, p < 0.001) and BS (M = 1.92%, SD = 1.2%, Z = -3.724, p < 0.001). There was no significant difference between CP and BS ($p = 0.7907$).

Speed. Significant difference was found ($F_{2,34} = 22.88, p < 0.001$). The typing speed under CP (M = 13.7 WPM, SD = 1.8) was significantly higher than that under IT (M = 11.4 WPM, SD = 1.6, p < 0.001) and BS (M = 11.7 WPM, SD = 1.3, p < 0.001). There was no significant difference between IT and BS ($p = 0.7606$).

Substitution number. We found significant difference ($\chi^2(2) = 27.758, p < 0.001$). The substitution number under IT (M = 45.3, SD = 35.4) was significantly larger than that under CP (M = 12, SD = 10, Z = -3.622, p < 0.001) and BS (M = 15.3, SD = 14.6, Z = -3.621, p < 0.001). There was no significant difference between CP and BS ($Z = -1.621, p = 0.105$).

Insertion number. We found significant difference ($\chi^2(2) = 26, p < 0.001$). The insertion number under IT (M = 26.6, SD = 33) was significantly larger than that under CP (M = 3.2, SD = 2.3, Z = -3.517, p < 0.001) and BS (M = 3.8, SD = 4.1, Z = -3.622, p < 0.001). There was no significant difference between the CP and BS ($Z = -0.281, p = 0.779$).

Omission number. There was no significant difference ($\chi^2(2) = 1.458, p = 0.482$) among IT (M = 3.4, SD = 3.0), CP (M = 2.3, SD = 1.2), and BS (M = 2.5, SD = 1.9).

Penetration number. Significant difference between IT and BS was found ($Z = -3.724, p < 0.001$). The penetration number under IT (M = 1135.5, SD = 38.1) was significantly larger than that under BS (M = 296.8, SD = 162.9).

Penetration depth. Significant difference between IT and BS was found ($Z = -3.920, p < 0.001$). The penetration depth
under IT (M = 10.1 mm, SD = 1.9) was significantly larger than that under BS (M = 1.6 mm, SD = 1.1).

Workload. We found significant difference ($\chi^2(2) = 21.6, p < 0.001$). The workload under IT (M = 10.2, SD = 3.2) was significantly larger than that under CP (M = 6.1, SD = 3, Z = -3.44, p < 0.01) and BS (M = 7.5, SD = 2.6, Z = -3.43, p < 0.01). The workload under BS was significantly larger than that under CP (Z = -2.79, $p = 0.006$).

User experience. When asked to compare BS and IT, all participants preferred BS over IT. One participant’s comment was representative, “Under IT, I often mis-tapped another button adjacent to my target button. When I moved my finger from one target button to another, I had to be very careful to keep a safe distance between my finger and the keyboard, or my finger might ‘scratch’ a button between the two target buttons. In contrast, the BS was much thinner than a finger, thus, mis-tapping seldom happened. Also, the ‘shatter’ and ‘recover’ visual cues guaranteed that I could keep a safe distance well and I did not need to worry the scratching problem.” When asked to compare BS and CP, most participants (17/18) commented that BS was a promising alternative to CP in some common scenarios. One participant said, “Because raising an arm in mid-air is more tired than holding a controller, I want to use CP if a controller is available at home/office. However, I do not want to carry a controller when I am outside. For example, if I want to use my VR headset to watch a movie on the plane, I hope to carry only my headset because the room of my suitcase is always not that enough. In this scenario, BS is a good alternative.” One participant provided another VR/AR usage scenario, “If I wear an AR headset to go shopping, BS is more suitable because I need text entry only when I reply to short immediate messages. My hands are mainly used for grabbing commodities. Grabbing a controller will destroy my shopping experience.” One participant said, “when I am interacting with the virtual world using bare hands, I prefer to reply immediate messages using BS. Because immediate messages are short, I am really reluctant to find my controller.”

**Fig. 5.** The experiment results, including uncorrected error rate, speed, substitution number, insertion number, omission number, penetration number, penetration depth, and workload. IT, CP, and BS are abbreviations for isomorphic typing, controller pointing, and the BrittleStylus technique. * means $p < 0.05$; ** means $p < 0.01$; *** means $p < 0.001$.

**Fig. 6.** The performance trends across three consecutive days.
Besides, one participant suggested that, “now you only use ‘disappear’ to represent ‘shatter.’ It is kind of abstract. If you render fragments of the stylus at the moment it shatters, the visual effect of ‘shatter’ is more realistic.”

Performance trends across the three experimental days. As shown in Fig. 6, it seemed that the participants could become proficient in using BS and CP quickly on the first day because the differences between the error rates across the three days under BS and CP are small. In contrast, participants could not type with bare fingers proficiently on the first day; though still higher than 5%, the error rate dropped obviously in the following two days. For the typing speed, all techniques shared a similar trend that the typing speed increased across the three days. The penetration number and distances under BS and IT are basically stable across the three days.

Summary. The BrittleStylus technique could enhance the typing accuracy of naive isomorphic typing significantly by 71% without compromising typing speed. By adopting the BrittleStylus technique, the substitution number and insertion number were significantly reduced by 30 (from 43.5 to 15.3) and 22.8 (from 26.6 to 3.8), while the omission number was not significantly reduced by 0.9 (from 3.4 to 2.5). As the reductions in the substitution and insertion numbers were attributed to using a thin stylus and adopting the “approach-transform-recover” mechanism, respectively, the visual cues provided by the “approach-transform-recover” mechanism contributed ~42% (= 22.8/(30+22.8+0.9)) of the total reduction, which was significant. Also, participants preferred BS over CP when they are outside or the text to be typed is too short to grab a controller during freehand interactions.

8 USER STUDY 2

There was a color difference in our experiment design which we previously neglected in Section 7. We conducted a follow-up user study to explore whether the color of a virtual tool (BrittleStylus/controller’s ray) affects the experiment results.

Participants. We recruited 8 participants (all males) aged between 22-26 (mean: 23.8) from campus. They were all right-handed.

Apparatus and Task. The hardware setup, the virtual experimental environment, and the task were the same as those in Section 7.

Experimental Design. The levels of the virtual tool color factor were red and purple, while those of the virtual tool factor were BrittleStylus and controller’s ray. The orders of the levels of the two factors were counterbalanced across the participants. Participants were asked to transcribe fifteen short sentences (each consists of 20-25 characters). The total length was 379. All conditions (combinations of the two factors) shared the same set of sentences. The dependent variables were: (i) uncorrected error rate; and (ii) speed.

Experiment Results. After checking normality and homogeneity of variance, we found that uncorrected error rate in this user study was not normally distributed. We used Aligned Rank Transform (ART) [63] to analyze uncorrected error rate. We adopted two-way ANOVA to analyze speed.

Uncorrected Error Rate. The results did not show a significant interaction between the virtual tool color factor and virtual tool factor ($F_{1,7} = 0.266, p = 0.622$). There was no significant difference between levels of the virtual tool color factor ($F_{1,7} = 3.718, p = 0.095$). There was also no significant difference between levels of virtual tool factor ($F_{1,7} < 0.001, p = 1.000$).

Speed. The results did not show a significant interaction between the virtual tool color factor and virtual tool factor ($F_{1,28} = 0.055, p = 0.8162$). There was no significant difference between levels of the virtual tool color factor ($F_{1,28} = 0.134, p = 0.7176$). There was also no significant difference between levels of virtual tool factor ($F_{1,28} = 3.086, p = 0.0899$).

Summary. We could not find any trends showing the color of virtual tools affected the typing performance. All participants also commented that they felt that the color of the virtual tool did not affect their performance. In addition, the comparison between BS and CP on performance was similar to that in Section 7.
The Experiment Results, Including Uncorrected Error Rate, Speed, Penetration Number and Penetration Depth

| Technique | Uncorrected Error Rate Mean versus RBS | Technique | Speed Mean (WPM) versus RBS | Penetration Number Mean versus RBS | Penetration Depth Mean versus RBS |
|-----------|----------------------------------------|-----------|-----------------------------|----------------------------------|----------------------------------|
| RBS 2.1%  | RBS 11.73                              | CH 11.72  | p = 0.737                   | UH 346.1 p < 0.0001              | UH 9.17 p < 0.0001               |
| 2H 3.8%   | p < 0.01                               | CH 11.72  | p = 0.737                   | CH 354.75 p < 0.0001             | CH 10.27 p < 0.0001              |
| CH 4.2%   | p < 0.001                              | OC 11.7   | p = 0.55                    | CH 354.75 p < 0.0001             | CH 10.27 p < 0.0001              |
| OC 4.5%   | p < 0.001                              | CH 11.6   | p = 0.794                   | OC 356.2 p < 0.0001              | OC 10.73 p < 0.0001              |

- **Constrained hand + unconstrained hand (2H),** i.e., after the virtual hand is constrained by a virtual button, an unconstrained mesh hand that represents the user’s real hand was rendered (see Fig. 7c); and
- **RBS**, the Realistic version of BrittleStylus technique. Besides the design of the BrittleStylus technique, the stylus shatters into dozens of fragments after contacting the virtual keyboard (see Fig. 7d), which was suggested by P15 in user study 1.

**Participants.** We recruited 20 participants (16 males and 4 females) aged between 20-26 (mean: 23.0) from campus. They are all right-handed. Each participant signed an informed consent form before the user study.

**Apparatus.** The hardware setup and the virtual environmental were the same as those in Section 5. We developed CH, OC, 2H, and RBS as follows.

- **CH.** Upon our system detected that the index finger of an avatar hand contacted a virtual button, the pose of the avatar hand at this moment was recorded. Then our system made the avatar hand invisible and rendered its 3D model with the recorded pose.
- **OC.** Following Prachyabrud and Borst’s work [7], we defined a normalized penetration depth \( p_{\text{norm}} \), which was \( \min(p/\text{P}, 1) \) where \( \text{P} \) was 10 mm, the mean penetration depth of participants in our user study 1; then the button color was calculated by \( \text{NC} + (\text{RC} - \text{NC}) * p_{\text{norm}} \), where \( \text{NC} \) was the normal button color, i.e., white, while RC was red.
- **2H.** To render a mesh hand that represents a real hand, our system acquired the vertices of the model of the invisible avatar hand associated with the real hand, connected adjacent vertices with line segments, and rendered them.
- **RBS.** We adopted the particle system component of Unity engine to render the realistic visual effect of “shattering into fragments.” The parameters of RBS were the same as those in user study 1.

If a button was successfully triggered, the system would still play a click sound as acoustic feedback under all conditions. Also, the system made the button grey for 200 ms as visual feedback under all conditions except OC. Under OC, the button directly turned red after the touch.

**Experimental Design.** Each participant performed the task described in Section 7 under the above five conditions. The order of the conditions was counterbalanced across the participants using a balanced Latin square design. Participants were asked to transcribe fifteen short sentences (each consists of 20-25 characters). The total length was 374. All five conditions shared the same set of sentences. The transparency level of the avatar hand was set to 0.51 by referring to Prachyabrud et al.’s work [7]. We still collected the following quantitative data: (i) uncorrected error rate; (ii) speed; (iii) penetration number; and (iv) penetration depth. The whole experiment took around 45 minutes. After the experiment, we asked participants to rate on the NASA-TLX form [62] for workload, \( E^2 \) [64] for immersion and enjoyment, a questionnaire adapted from Rietzler et al. [1]’s work, and the overall preference. The questionnaire included six statements on a 7-point Likert scale (1: strongly disagree, 7: strongly agree) as follows: (i) I had the feeling of touching the virtual buttons (touch); (ii) I could feel the resistance of virtual buttons when I was typing (resistance); (iii) The representation of physical constraints felt realistic (realistic); (iv) The representation of physical constraints was sufficient for isomorphic typing (sufficient); (v) I liked the representation of physical constraints (like); and (vi) I felt the representation of physical constraints was disturbing (disturb). Also, the experimenter talked with each participant and recorded their comments on their experience during the user study.

**Experiment Results.** Tables 1, 2, 3, and 4 show the results. After checking normality and homogeneity of variance, we found that speed and penetration depth were not normally distributed; uncorrected error rate and penetration depth’s variances were not homogeneous. Then we adopted the non-parametric Friedman tests with post hoc Wilcoxon Signed Rank tests to analyze all the dependent variables. For pairwise comparisons, we only report those with significance (\( p < 0.05 \)).

**Uncorrected Error Rate.** Significant difference between the five techniques was found (\( \chi^2(4) = 21.071, p < 0.001 \)). The uncorrected error rate under RBS (M = 2.11%, SD = 1.0%) was significantly lower than that under 2H (M = 3.8%, SD = 1.76%, \( Z = -3.043, p < 0.01 \)), CH (M = 3.9%, SD = 1.57%, \( Z = -3.136, p < 0.01 \)), CH (M = 4.2%, SD = 1.71%, \( Z = -3.324, p < 0.001 \)), and OC (M = 4.48%, SD = 1.90%, \( Z = -3.381, p < 0.001 \)). The uncorrected error rate under 2H was significantly lower than that under OC (\( Z = -2.036, p < 0.05 \)).

**Penetration Number.** Significant difference was found (\( \chi^2(4) = 45.571, p < 0.001 \)). The penetration number under RBS (M = 59.1, SD = 34.7) was significantly lower than that under UH (M = 346.1, SD = 15.1, \( Z = -3.92, p < 0.0001 \)), 2H (M = 350.9, SD = 18.0, \( Z = -3.92, p < 0.0001 \)), CH (M = 354.8, SD = 14.4, \( Z = -3.92, p < 0.0001 \)), and OC (M = 356.2, SD = 17.2, \( Z = -3.92, p < 0.0001 \)). The penetration number under...
We found no significant difference (p > 0.05). The penetration depth under UH (M = 1.6 mm, SD = 3.6, Z = -3.920, p < 0.0001), CH (M = 10.3 mm, SD = 3.4, Z = -3.920, p < 0.0001), and OC (M = 10.7 mm, SD = 3.3, Z = -3.920, p < 0.0001). The penetration depth under UH was significantly lower than that of OC (Z = -1.979, p < 0.05).

Workload. We found significant difference (χ²(4) = 18.22, p < 0.01). The workload score under RBS (M = 6.3, SD = 3.4) was significantly smaller than that under UH (M = 8.2, SD = 3.4, Z = -3.06, p < 0.01), OC (M = 8.7, SD = 3.7, Z = -3.47, p < 0.001), CH (M = 8.85, SD = 3.1, Z = -3.55, p < 0.001) and 2H (M = 10.25, SD = 4.1, Z = -3.55, p < 0.001). The workload score under 2H was significantly larger than that under CH (Z = -2.178, p < 0.05) and UH (Z = -2.335, p < 0.05).

Immersion. We found significant difference (χ²(4) = 21.64, p < 0.001). The immersion score under RBS (M = 5.25, SD = 0.9) was significantly smaller than that under CH (M = 4.65, SD = 0.7, Z = -2.96, p < 0.01), 2H (M = 4.55, SD = 0.8, Z = -2.67, p < 0.01), and OC (M = 4.55, SD = 0.8, Z = -3.58, p < 0.001), and UH (M = 4.5, SD = 0.8, Z = -3.59, p < 0.001).

Enjoyment. We found significant difference (χ²(4) = 24.21, p < 0.001). The enjoyment score under RBS (M = 6, SD = 1.3) was significantly larger than that under UH (M = 4.6, SD = 1.4, Z = -2.87, p < 0.01), OC (M = 4.45, SD = 1.2, Z = -3.29, p < 0.01), 2H (M = 4.35, SD = 1.6, Z = -2.67, p < 0.01), and CH (M = 4.2, SD = 1.0, Z = -3.49, p < 0.001).

Touch. We found significant difference (χ²(4) = 10.7, p < 0.05). The touch score under RBS (M = 5.7, SD = 1.1) was significantly larger than that under CH (M = 4.9, SD = 1.1, Z = -2.76, p < 0.01), OC (M = 4.8, SD = 1.5, Z = -2.39, p < 0.05), UH (M = 4.55, SD = 1.7, Z = -2.27, p < 0.05) and 2H (M = 4.5, SD = 1.4, Z = -2.84, p < 0.01).

Resistance. We found significant difference (χ²(4) = 19.36, p < 0.001). The resistance score under UH (M = 3.45, SD = 1.4) was significantly smaller than that under CH (M = 5.15, SD = 1.4, Z = -3.192, p < 0.01), RBS (M = 5.1, SD = 1.6, Z = -3.24, p < 0.01), 2H (M = 5, SD = 1.7, Z = -2.96, p < 0.01), and OC (M = 4.9, SD = 1.6, Z = -3.095, p < 0.01).

Realistic. We found no significant difference (χ²(4) = 7.0, p = 0.14).

Sufficient. We found significant difference (χ²(4) = 22.47, p < 0.01). The sufficient score under RBS (M = 6.05, SD = 0.9) was significantly larger than that under CH (M = 4.85, SD = 1.2, Z = -2.94, p < 0.01), UH (M = 4.75, SD = 1.2, Z = -3.21, p < 0.01), CH (M = 4.5, SD = 0.9, Z = -3.34, p < 0.001), and 2H (M = 4.3, SD = 1.6, Z = -3.09, p < 0.01). The sufficient score under OC was significantly larger than that under 2H (Z = -1.99, p < 0.05).

Like. We found significant difference (χ²(4) = 20.16, p < 0.01). The like score under RBS (M = 5.7, SD = 1.6) was significantly larger than that under UH (M = 4.95, SD = 1.2, Z = -2.14, p < 0.05), OC (M = 4.5, SD = 1.2, Z = -3.33, p < 0.001), CH (M = 4.5, SD = 0.9, Z = -2.56, p < 0.05), and 2H (M = 4.25, SD = 1.7, Z = -2.61, p < 0.01).

---

**TABLE 2**

The Experiment Results, Including the Score of Workload, Immersion, Enjoyment and Touch

| Technique | Mean versus RBS | Mean versus RBS | Technique | Mean versus RBS | Mean versus RBS |
|-----------|-----------------|-----------------|-----------|-----------------|-----------------|
| RBS       | 6.3             | 5.25            | RBS       | 6.05            | 5.7             |
| UH        | 8.2 p < 0.01    | CH 4.65 p < 0.01| OC 4.45 p < 0.01 |
| OC        | 8.7 p < 0.001   | 2H 4.55 p < 0.01| 2H 4.35 p < 0.01 |
| CH        | 8.85 p < 0.001  | OC 4.55 p < 0.001| CH 4.2 p < 0.001 |
| 2H        | 10.25 p < 0.001 | UH 4.5 p < 0.001| 2H 4.5 p < 0.01  |

**TABLE 3**

The Experiment Results, Including the Score of Resistance, Realistic, Sufficient and Like

| Technique | Mean versus RBS | Mean versus RBS | Technique | Mean versus RBS | Mean versus RBS |
|-----------|-----------------|-----------------|-----------|-----------------|-----------------|
| RBS       | 6.05            | 5.7             | RBS       | 6.05            | 5.7             |
| OC        | 4.85 p < 0.01   | CH 4.95 p < 0.01| OC 4.5 p < 0.001 |
| 2H        | 5 p < 0.01      | UH 4.9 p < 0.001| 2H 4.5 p < 0.05  |
| CH        | 4.9 p < 0.01    | OC 4.8 p < 0.001| CH 4.75 p < 0.001| |
| UH        | 3.45 p < 0.01   | 2H 4.6 p < 0.001| OC 4.5 p < 0.01  |

**TABLE 4**

The Experiment Results, Including the Score of Disturb and Overall Preference

| Technique | Mean versus RBS | Mean versus RBS |
|-----------|-----------------|-----------------|
| RBS       | 3.05            | CH 4.65 p < 0.001 |
| UH        | 3.15 p < 0.793  | OC 4.45 p < 0.001 |
| OC        | 3.75 p < 0.112  | CH 4.35 p < 0.001 |
| CH        | 3.9 p < 0.05    | 2H 4.3 p < 0.01  |
| 2H        | 4.55 p < 0.01   | OC 4.3 p < 0.001 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We found significant difference ($\chi^2(4) = 11.08, p < 0.05$). The disturb score under RBS (M = 3.05, SD = 1.8) was significantly smaller than that under CH (M = 3.9, SD = 1.4, Z = -2.30, p < 0.05) and 2H (M = 4.55, SD = 1.5, Z = -6.21, p < 0.01). The disturb score under UH (M = 3.15, SD = 1.6) was significantly smaller than that under CH (Z = -2.546, p < 0.05) and 2H (Z = -2.58, p < 0.01). The disturb score under OC (M = 3.75, SD = 1.4) was significantly smaller than 2H (Z = -2.389, p < 0.05).

Overall Preference. We found significant difference ($\chi^2(4) = 29.92, p < 0.001$). The overall preference score under RBS (M = 6.4, SD = 0.8) was significantly larger than that under CH (M = 4.65, SD = 1.1, Z = -3.35, p < 0.001), CH (M = 4.45, SD = 1.3, Z = -3.53, p < 0.001), 2H (M = 4.35, SD = 1.7, Z = -3.38, p < 0.001), and OC (M = 4.3, SD = 1.5, Z = -3.71, p < 0.001).

User Experience. All participants said that the degrees of the illusion of kinesthetic feedback induced by CH, 2H, OC, and RBS were similar. Most participants (18/20) commented that the typing experience under RBS was fluent while that under all three pseudo-haptic (+ visual cue) conditions (CH, OC, 2H) was not. Twelve participants mentioned the keywords “lag” or “stutter” when describing the typing experience under the above three conditions. One participant’s comment was representative, “Under CH, OC, and 2H, the movement of my avatar hand stuttered each time it contacted the keyboard, which made me feel bad. Under RBS, the visual effect of the stylus shattering into fragments was cool. And the movement of the avatar hand was fluent, which was an enjoyable experience.” Eight participants mentioned that they did not pay much attention to the visual cue (mesh hand and color changing) under OC and 2H. Five commented that the illusion of resistance under CH, OC, and 2H conditions seemed unable to reduce the error rate, while RBS helped them type accurately.

Discussion

For enhancing the participants’ typing performance, RBS was as effective as its abstract version in user study 1, while the pseudo-haptic + visual cue methods (OC and 2H) were not. Under CH, the penetration depth in grasping 3D virtual objects was ~34mm [7], which was three times higher than that in isomorphic typing (10mm) in this experiment. This comparison suggested that users tend to grasp a virtual 3D object “tightly” and “steadily” to manipulate it, while they tend to just tap a virtual button “slightly” to select it. In other words, users are likely to regard grasping as a more complex task and tapping as a simpler one. Thus, under OC and 2H, users might think that it is worth to pay attention to the visual cues when grasping, while they might not think so when tapping. We believe this is the reason why OC and 2H are effective in enhancing the performance of grasping [7] but not effective in enhancing that of isomorphic typing. Unlike users can still trigger buttons while neglecting the visual cues under 2H and OC, the “shatter-recover” mechanism of RBS re-shapes the hand movement of isomorphic typing, thus users have to “take the ‘recover’ visual cue seriously,” or they can not trigger buttons.

To induce the illusion of kinesthetic feedback, RBS constrains the appendage rather than the avatar hand, which is an indirect way, while CH, OC, and 2H directly constrain the avatar hand. The resistance score showed that the degree of kinesthetic feedback indirectly induced by RBS was parity with that directly induced by CH, OC, or 2H. This result is consistent with the argument that tools become extensions of our bodies after we use them proficiently [54]. Also, the touch score under RBS was significantly larger than that under UH while those under the pseudo-haptic (+ visual cue) methods were not. Under the pseudo-haptic (+ visual cue) methods, after the very moment the tip of an avatar finger touches a virtual button, the real fingertip is actually in the space behind the button, i.e., the proprioception does not match the vision. Though the user sees the avatar fingertip is touching the virtual button like its counterpart in the real world, her brain unlikely believe so due to the above mismatch. In contrast, apart from that the proprioception matches the vision under RBS, the visual effect of shattering into fragments “underscores” the touch event.

As isomorphic typing consists of repetitive fast-paced hand movement, users are very sensitive to the fluency of hand movement in this experiment. We believe that this resulted in that participants’ ratings on several metrics (workload, immersion, enjoyment, sufficient, like) for RBS significantly better than those for the CH, 2H, and OC conditions.

Summary

Compared to pseudo-haptic (+ visual cue) methods, kine-appendage is more suitable for enhancing the performance and subjective experience of isomorphic typing.

10 OVERALL DISCUSSION AND FUTURE WORK

Kine-appendage & users’ view direction. Theoretically, if a kine-appendage is parallel to users’ view direction, they may not see its transformations well. Nevertheless, users mainly use the tips of kine-appendages for interaction (e.g., pressing, poking, grabbing, etc.), then they want to see the tips of the kine-appendages, making the appendages not parallel to their view direction. Also, current commodity VR/AR headsets (Oculus quest, HTC cosmos, HoloLens 2, etc.) utilized their front-facing cameras to track fingers. If a finger (sharing the same line with the appendage initially) is parallel to users’ view direction, the VR/AR headsets cannot track the finger, not to mention render the appendage.

Adding Texture to the Kine-Appendages. As we found in the pilot study 1, participants could not see the other three transformations (“hinge”, “bend”, and “break”) well when their view direction was nearly parallel to the plane determined by the stylus’s centerline that rotates, bends, or breaks (see the bend transformation in Fig. 2). We believe that one primary reason is that the appendage has no texture. If we add rich texture to the appendage, e.g., we render the appendage as a wooden stick, users may better perceive the deformation of the appendage from their egocentric view by seeing the twisted texture of the appendage. In the future, we will investigate the effects of putting textures onto the appendage.

The Role of Auditory Feedback. We have provided auditory feedback for the pressing interaction under all conditions in all user studies. In the future, we will investigate whether auditory feedback helps the pressing interaction substantially. We will also explore using the frequency and
amplitude of peeping sounds to indicate the distance between the tip of a finger or an appendage and the virtual keyboard.

Exploring More Transformations of the Kine-Appendage in More Application Scenarios. Although “shatter” beat the other three visual transformations introduced in Section 1, the other three visual transformations could be suitable for at least the following three scenarios. First, for the grabbing-releasing scenario, we can append two styluses to the index fingertip and thumb fingertip of the avatar hand respectively, and encode the interaction status into the color of the styluses that rotate to guide users (see HingeStylus in Fig. 8). Due to the lack of real physical constraints, the real fingers sink into a virtual object when a user grabs it under UH, leading to the problem of “sticking object.” This problem refers to the fact that when the user releases a grabbed virtual object, exaggerated finger motions are required, degrading release performance and subjective experience and contributing to fatigue [7]. Users were recommended to touch virtual objects lightly when grabbing them to alleviate this problem. Though OC and 2H give visual cues to help users to achieve “light touch” more easily, the real fingers still sink into the virtual object. In contrast, HingeStylus has the potential to solve the “sticking object” problem completely, as users can keep their hand out of the virtual object for the whole grabbing-releasing period.

Second, the appended stylus can also be used for displaying different masses of virtual objects when they are pushed. The heavier a virtual object is, the larger the friction of the supporting surface is, the stronger the force needs to be exerted to move the object, then the stylus bends more under a stronger force (see ToughStylus in Fig. 9). Third, a wall’s strong resistance can be displayed by making the ToughStylus “break” after the user’s pushing force exceeds a threshold (see another version of ToughStylus in Fig. 10).

Translation of the Virtual Appendage? The transformation of an object includes rigid transformations, i.e., translation and rotation, and non-rigid transformation, i.e., deformation. By far, we have proposed utilizing the deformation and rotation of the virtual appendage to guide users during freehand interaction. We also considered utilizing the translation of the virtual appendage, which means that both ends of the virtual appendage are not always attached to the avatar hand, and the appendage can be translated in the proximity of the avatar hand. However, we are not sure whether users still regard an appendage that is not attached to the avatar hand as an extension of their bodies or not, which might make it difficult to induce the illusion of kinesthetic feedback. We will investigate the effects of the translation of the virtual appendage in the future.

11 LIMITATIONS OF BRITTLESTYLOS

Next, we discuss the limitations of BrittleStylus.

The parameters of BrittleStylus, i.e., length, diameter, and threshold distance, were determined heuristically by averaging the preferred values from several participants. They can be used as the default values if we do not know

---

**Fig. 8.** The HingeStylus technique (potential future work) is based on the “hinge” transformation introduced in Section 1. It is designed for enhancing the experience of grabbing and releasing virtual objects with bare hands. The two red styluses hinged to the tips of the thumb and index fingers are HingeStyluses. We denote the angle between the first segment of an index/thumb finger and a HingeStylus as “hinge angle.” (a) When the user’s hand is approaching a virtual object, the hinge angles are 180°. (b) After a stylus contacts the virtual object, the hinge angle becomes smaller and smaller. (c) When both hinge angles are smaller than the threshold angle, the virtual object is grabbed, and the color of HingeStyluses changes. (d) The user extends her fingers until the hinge angles become 180°, then the virtual object is released.

**Fig. 9.** The ToughStylus technique (potential future work) is based on the “bend” transformation introduced in Section 1. It is designed for displaying different masses of virtual objects when a user pushes them using an appended stylus. The heavier a virtual object is, the user needs to exert a stronger force to make the object move, then the stylus bends more under a stronger force. See the different degrees of bending of the stylus when pushing a 5 kg cube (a) and a 10 kg cube (b).

**Fig. 10.** Another version of ToughStylus technique (potential future work) is based on the combination of the “bend” and the “break” transformations introduced in Section 1. It is designed for inducing the illusion of pushing a real wall. (a) A user pushes a wall using the appended stylus and it bends under the user’s force and the wall’s resistance. (b) The stylus breaks after the user’s force reaches a threshold.
anything about the current user. We expect users’ performance can be further enhanced if these parameters are tailored particularly to their characteristics. In the future, we hope to explore reinforced learning methods to dynamically set these parameters to fit users’ characteristics.

Dudley et al. [50] have found that, when typing with bare hands in mid-air in VR, the performance under the “two index fingers” condition (2-F) was significantly better than that under the “ten fingers” condition (10-F). The reason might be that using ten fingers to type without haptic feedback brings too much mental demand. Considering mental demand and gorilla-arm effect, we cannot clearly speculate which is the better one between 1-F (“one index finger” condition) and 2-F. Also, we could not find any work comparing 1-F and 2-F in VR to our knowledge. Thus, we should have considered both 1-F and 2-F when enhancing isomorphic typing with kine-appendage, which is also a limitation of this work. In the future, we will compare 1-F and 2-F through an empirical study. Despite this, 1-F itself is essential from the perspective of practical usage. First, we must consider the typing needs of disabled VR users with unilateral upper limb disability. Second, when we use BrittleStylus in AR scenarios in the future, we must note that AR users often have only one hand available for typing when they are not at home or office, e.g., they have to use one hand to grab a handrail on a bus/subway, they use one hand to lift a bag on their way back after shopping, etc.

12 Conclusion

In this work, we propose the kine-appendage concept, i.e., a virtual appendage is added to the user’s avatar hand; and the appendage exhibits transformations as it contacts a virtual object and recovers its original appearance as it disengages from the contact. Then we introduced four example kine-appendage virtual transformations: “shatter,” “hinge,” “bend,” and “break.” Further, we designed a proof-of-concept kine-appendage technique, BrittleStylus based on the “shatter” transformation and evaluated it through three empirical studies. The main results are (i) the BrittleStylus technique reduced the uncorrected error rate of naive isomorphic typing significantly from 6.53% to 1.92% without compromising typing speed; (ii) the BrittleStylus technique could induce an illusion of kinesthetic feedback, the degree of which was parity with that induced by pseudo-haptic (+ visual cue) methods; and (iii) participants were very sensitive to the fluency of hand movements in isomorphic typing. BrittleStylus beat pseudo-haptic (+ visual cue) methods because of not only good performance but also fluent hand movements. This work was approved by the ethics committee of Guangxi University (reference number: GXU-2021-156).

References

[1] M. Rietzler, F. Geiselhart, J. Frommel, and E. Rukzio, “Conveying the perception of kinesthetic feedback in virtual reality using state-of-the-art hardware,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2018, pp. 1–13.
[2] J. Blake and H. B. Gurocak, “Haptic glove with MR brakes for virtual reality,” IEEE/ASME Trans. Mechatron., vol. 14, no. 5, pp. 606–615, Oct. 2009.
[3] D. Tsetsesukou, K. Sato, and S. Tachi, “Exointerfaces: Novel exoskeleton haptic interfaces for virtual reality, augmented sport and rehabilitation,” in Proc. 1st Augmented Hum. Int. Conf., 2010, pp. 1–6.
[4] P. Lopes, S. You, L.-P. Cheng, S. Marwecki, and P. Baudisch, Proving Haptics to Walls & Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation. New York, NY, USA: ACM, 2017, pp. 1471–1482.
[5] M. Pfeiffer, T. Dünne, S. Schneeegg, F. Alt, and M. Rohs, “Cruise control for pedestrians: Controlling walking direction using electrical muscle stimulation,” in Proc. 33rd Ann. ACM Conf. Hum. Factors Comput. Syst., 2015, pp. 2505–2514.
[6] A. Lécuyer, “Simulating haptic feedback using vision: A survey of research and applications of pseudo-haptic feedback,” Presence, vol. 18, no. 1, pp. 39–53, 2009.
[7] M. Prachyabrued and C. W. Borst, “Design and evaluation of visual interpenetration cues in virtual grasping,” IEEE Trans. Vis. Comput. Graph., vol. 22, no. 6, pp. 1718–1731, Jun. 2016.
[8] R. Canales, A. Normoyle, Y. Sun, Y. Ye, M. D. Luca, and S. Jörg, “Virtual grasping feedback and virtual hand ownership,” in Proc. ACM Symp. Appl. Percept., 2009, pp. 1–9.
[9] Y. Tian, Y. Bai, S. Zhao, C.-W. Fu, T. Yang, and P. A. Heng, “Virtually-extended proprioception: Providing spatial reference in VR through an appended virtual limb,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2020, pp. 1–12.
[10] H. Hoffmann, “Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments,” in Proc. Virtual Reality Ann. Int. Conf., 2015, pp. 127–134.
[11] A. L. Simeone, E. Velloso, and H. Gellersen, “Substitutional reality: Using the physical environment to design virtual reality experiences,” in Proc. 33rd Ann. ACM Conf. Hum. Factors Comput. Syst., 2015, pp. 3307–3316.
[12] M. Sra, S. Garrido-Jurado, C. Schmandt, and P. Maes, “Procedurally generated virtual reality from 3D reconstructed physical space,” in Proc. 22nd ACM Conf. Virtual Reality Softw. Technol., 2016, pp. 191–200.
[13] L. Shapira and D. Freedman, “Reality skins: Creating immersive and tactile virtual environments,” in Proc. IEEE Int. Symp. Mixed Augmented Reality, 2016, pp. 115–124.
[14] P. E. Gruenbaum, W. A. McNeely, H. A. Sowizral, T. L. Overman, and B. W. Knutson, “Implementation of dynamic robotic graphics for a virtual control panel,” Presence Teleoperators Virtual Environ., vol. 6, no. 1, pp. 118–126, 1997.
[15] W. McNeely, “Robotic graphics: A new approach to force feedback for virtual reality,” in Proc. IEEE Virtual Reality Ann. Int. Symp., 1993, pp. 336–341.
[16] Y. Yokokohji, Y. Sugawara, J. Kinoshita, and T. Yoshikawa, “Mechano-media that transmit kinesthetic knowledge from a human to other humans,” in Robotics Research. Berlin, Germany: Springer, 2003, pp. 499–512.
[17] L.-P. Cheng et al., “Turkdeck: Physical virtual reality based on people,” in Proc. 28th Ann. ACM Symp. User Interface Softw. Technol., 2015, pp. 417–426.
[18] L. Kohli, “Redirected touching: Warping space to remap passive haptics,” in Proc. IEEE Symp. 3D User Interfaces, 2010, pp. 129–130.
[19] M. Azmandian, M. Hancock, H. Benko, E. Olek, and A. D. Wilson, “Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2016, pp. 1668–1679.
[20] M. Bouzit, G. Burdea, G. Popescu, and R. Boian, “The rutgers master ii-new design force-feedback glove,” IEEE/ASME Trans. Mechatron., vol. 7, no. 2, pp. 256–263, Jun. 2002.
[21] T. Endo et al., “Five-fingered haptic interface robot: Hiro III,” in Proc. 3rd. Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environment, Teleoperator Syst., 2009, pp. 458–463.
[22] J. C. Perry, J. Rosen, and S. Burns, “Upper-limb powered exoskeleton design,” IEEE/ASME Trans. Mechatron., vol. 12, no. 4, pp. 408–417, Aug. 2007.
[23] A. Kron and G. Schmidt, “Multi-fingered tactile feedback from virtual and remote environments,” in Proc. 11th Symp. Haptic Interface for Virtual Environments and Robotics, 2003, pp. 16–23.
[24] F. Farbiz, Z. H. Yu, C. Manders, and W. Ahmad, “An electrical muscle stimulation haptic feedback for mixed reality tennis game,” in Proc. Conf. ACM SIGGRAPH, 2007, pp. 140–es.
[25] L. Shapira and D. Freedman, “Reality skins: Creating immersive and tactile virtual environments,” in Proc. IEEE Int. Symp. Mixed Augmented Reality, 2016, pp. 115–124.

Authorized licensed use limited to the terms of the applicable license agreement with JEEE. Restrictions apply.
P. Lopes, A. Ion, and P. Baudisch, “Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation,” in Proc. 28th Annu. ACM Symp. User Interface Softw. Technol., 2015, pp. 11–19.

E. Tamaki, T. Miyaki, and J. Rekimoto, “Possessedhand: Technique for controlling human hands using electrical muscles stimul,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2011, pp. 543–552.

L. Liu, C. Yao, Y. Liu, P. Wang, Y. Chen, and F. Ying, “FlowGlove: A liquid-based wearable device for haptic interaction in virtual reality,” in Proc. Int. Conf. Hum.-Comput. Interact., 2020, pp. 316–331.

T. M. Simon, R. T. Smith, and B. H. Thomas, “Wearable jamming methods for virtual environment haptics,” in Proc. ACM Int. Symp. Wearable Comput., 2014, pp. 67–70.

M. Rietzler, G. Haas, T. Dreja, F. Geiselhart, and E. Rukzio, “Virtual muscle force: Communicating kinesthetic forces through pseudo-haptic feedback and muscle input,” in Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol., 2019, pp. 931–922.

M. Rietzler, F. Geiselhart, J. Gügenheimer, and E. Rukzio, Breaking three-hand-tracking: Weight Perception Using Perceivable Tracking Offsets. New York, NY, USA: ACM, 2018, pp. 1–12.

Y. Ban and Y. Ujitoko, “Hit-stop in VR: Combination of pseudo-haptics and vibration enhances impact sensation,” in Proc. IEEE World Haptics Conf., 2021, pp. 991–996.

E.-C. Lee, Y.-H. Cho, and I.-K. Lee, “Simulating water resistance in a virtual underwater experience using a visual motion delay effect,” in Proc. IEEE Conf. Virtual Reality 3D User Interfaces, 2019, pp. 259–266.

H. Kang, G. Lee, and J. Han, “Visual manipulation for underwater drag force perception in immersive virtual environments,” in Proc. IEEE Conf. Virtual Reality 3D User Interfaces, 2019, pp. 38–46.

R. Yu and D. A. Bowman, “Pseudo-haptic display of mass and mass distribution during object rotation in virtual reality,” IEEE Trans. Vis. Comput. Graph., vol. 26, no. 5, pp. 2094–2103, May 2020.

T. J. Dube and A. S. Arif, “Text entry in virtual reality: A comprehensive review of the literature,” in Proc. Hum.-Comput. Interact. Recognit. Interaction Technol., 2019, pp. 419–437.

B. Jackson, L. B. Caraco, and Z. M. Spilka, “Arc-type and Tilt-type: Pen-based immersive text input for room-scale VR,” in Proc. Symp. Spatial User Interaction, 2020, pp. 1–10.

D.-M. Pham and W. Stuerzlinger, “Hawkey: Efficient and versatile text entry for virtual reality,” in Proc. 25th ACM Symp. Virtual Reality Softw. Technol., 2019, pp. 1–11.

H. Jiang and D. Weng, “HiPad: Text entry for head-mounted displays using circular touchpad,” in Proc. IEEE Conf. Virtual Reality 3D User Interfaces, 2020, pp. 692–703.

J. Adhikary and K. Vertanen, “Text entry in virtual environments using speech and a midair keyboard,” IEEE Trans. Vis. Comput. Graph., vol. 27, no. 5, pp. 2648–2658, May 2021.

Oculus Quest. Accessed: Nov. 25, 2021. [Online]. Available: https://www.oculus.com/quest-2/

HTC VIVE Pro. Accessed: Nov. 25, 2021. [Online]. Available: https://www.vive.com/hk/product/vive-pro/.

HTC VIVE hand tracking SDK. Accessed: Sep. 26, 2021. [Online]. Available: https://developer.vive.com/resources/knowledgebase/vive-hand-tracking-sdk/.

Leap motion. Accessed: Nov. 25, 2021. [Online]. Available: http://www.ultraleap.com/

T. Ni, D. Bowman, and C. North, “Airstroke: Bringing unistroke text entry to freehand gesture interfaces,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2011, pp. 2473–2476.

X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi, “ATK: Enabling ten-finger freehand typing in air based on 3D hand tracking data,” in Proc. 28th Annu. ACM Symp. User Interface Softw. Technol., 2015, pp. 539–548.

J. Fashimpaur, K. Kin, and M. Longest, “PinchType: Text entry for virtual and augmented reality using comfortable thumb to finger-tip pinches,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2020, pp. 1–11.

J. Speicher, A. M. Feit, P. Ziegler, and A. Krüger, “Selection-based text entry in virtual reality,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2018, pp. 1–13.

OptiTrack motion capture system. Accessed: Nov. 25, 2021. [Online]. Available: https://www.optitrack.com/

J. Dudley, H. Benko, D. Wigdor, and P. O. Kristensson, “Performance envelopes of virtual keyboard text input strategies in virtual reality,” in Proc. IEEE Int. Symp. Mixed Augmented Reality, 2019, pp. 289–300.

M. Richardson, M. Durasoﬀ, and R. Wang, “Decoding surface touch typing from hand-tracking,” in Proc. 33rd Annu. ACM Symp. User Interface Softw. Technol., 2020, pp. 686–696.

A. Gupta, M. Samad, K. Kin, P. O. Kristensson, and H. Benko, “Investigating remote tactile feedback for mid-air text-entry in virtual reality,” in Proc. IEEE Int. Symp. Mixed Augmented Reality, 2020, pp. 350–360.

D. Schneider et al., “Accuracy of commodity ﬁnger tracking systems for virtual reality head-mounted displays,” in Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops, 2020, pp. 804–805.

A. Alzayat, M. Hancock, and M. A. Nacenta, Quantitative Measurement of Tool Embodiment for Virtual Reality Input Alternatives. New York, NY, USA: ACM, 2019, pp. 1–11.

Oculus Integration SDK. Accessed: Nov. 25, 2021. [Online]. Available: https://developer.oculus.com/downloads/package/unify-integration/

D. Vogel and P. Baudisch, “Shift: A technique for operating pen-based interfaces using touch,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2007, pp. 657–666.

R. Balakrishnan and I. S. MacKenzie, “Performance differences in the fingers, wrist, and forearm in computer input control,” in Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst., 1997, pp. 303–310.

Gorilla Arm. Accessed: Nov. 25, 2021. [Online]. Available: http://cat.org/jargon/html/G/gorilla-arm.html

R. W. Soukoreﬀ and I. S. MacKenzie, “Metrics for text entry research: An evaluation of MSD and KSPC, and a new uniﬁed error metric,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2003, pp. 113–120.

Levenshtein distance. Accessed: Nov. 25, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Levenshtein_distance

H. Benko, A. D. Wilson, and P. Baudisch, “Precise selection techniques for multi-touch screens,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2006, pp. 1263–1272.

NASA TLX. Accessed: Aug. 28, 2021. [Online]. Available: https://humansystems.arc.nasa.gov/groups/tlx/

J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins, “The Levenshtein error metric,” in Proc. CHI Conf. Hum. Factors Comput. Syst., 2003, pp. 113–120.

J. W. Lin, H. Duh, D. Parker, H. Abi-Rached, and T. Furness, “Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment,” in Proc. Conf. IEEE Virtual Reality, 2002, pp. 164–171.

Yang Tian received the bachelor’s degree from Jilin University in 2013, and the MSc and PhD degrees from the Department of Computer Science and Engineering, The Chinese University of Hong Kong in 2014 and 2020, respectively. He is now an assistant professor with the Department of Computer Science and Technology, Guangxi University, China. His research interest is human computer interaction in VR/AR.

Hualong Bai received the bachelor’s degree from the Nanjing University of Information Science and Technology. He is currently working towards the master’s degree with the school of computers, electronics and information, Guangxi University, China.
Shengdong Zhao received the master's and PhD degrees from the University of California at Berkeley and the University of Toronto, respectively. He is currently an associate professor with the Department of Computer Science of the National University of Singapore where he founded the NUS-HCI research lab in 2009. He has a wealth of experience in developing new interface tools and applications. ACM Interaction Magazine mentioned him as one of the most prolific authors in Asia for the ACM SIGCHI conference. He frequently served on the program committees in top HCI conferences. He was the recipient of the ICACHI Outstanding Young Leaders in HCI Award and the NUS Young Investigator Award. More information about Shengdong can be found at http://www.shengdongzhao.com

Chi-Wing Fu (Member, IEEE) received the BSc degree in computer science from the Chinese University of Hong Kong and the PhD degree in computer science from Indiana University, Bloomington. He is a professor with the Department of Computer Science and Engineering, the Chinese University of Hong Kong and now serves as the associate dean with the Faculty of Engineering. Also, he served as the program co-chair of SIGGRAPH ASIA 2016 technical brief and poster, associate editor for Computer Graphics Forum and IEEE Computer Graphics and Applications, and program committee members in various conferences including SIGGRAPH, IEEE Visualization, etc. His research interests include computer graphics, 3D computer vision, and user interaction.

Chun Yu received the PhD degree from Tsinghua University in 2012. He is currently an associate professor with Tsinghua University. He is keen to research computational models and algorithms that can facilitate the development of high-efficiency multi-modal user interface, such as mobile phones, large displays, VR/AR headsets and so on. He has published more than 30 papers in prestigious conferences and journals in Human Computer Interaction, including CHI, UIST, CSCW and International Journal of Human-Computer Studies. His recent research won the ACM CHI 2017 Honorable Mention Award, ACM CHI 2016 Honorable Mention Award, ACM CHI 2015 Honorable Mention Award, ChineseCHI 2015 Best Paper Award, IEEE UIC 2014 Best Paper Award, ACM CHI 2013 Honorable Mention Award, ACM MobileHCI 2013 Honorable Mention Award. He was the recipient of Advances of Science and Technology (2015), First Prize, Administration of Education, China. He was the recipient of CCF-Intel Young Faculty Researcher Program in 2016 and CCF Young Talent Program in 2017.

Haozhao Qin received the bachelor's degree from the University of Shanghai for Science and Technology. He is currently working towards the master's degree with the school of computers, electronics and information, Guangxi University, China.

Qiong Wang is currently an associate researcher with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her research interests include VR applications in medicine, visualization, medical imaging, human–computer interaction, and computer graphics.

Pheng-Ann Heng (Senior Member, IEEE) received the BSc degree in computer science from the National University of Singapore, in 1985, and the MSc degree in computer science, the MArt degree in applied math, and the PhD degree in computer science, all from the Indiana University, in 1987, 1988, and 1992, respectively. He is a professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He has served as the department chairman from 2014 to 2017 and as the head of graduate division from 2005 to 2008 and then again from 2011 to 2016. He has served as the director of Virtual Reality, Visualization and Imaging Research Center, CUHK since 1999. He has served as the director of Center for Human-Computer Interaction, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences since 2006. He has been appointed by China Ministry of Education as a Cheung Kong scholar chair professor in 2007. His research interests include AI and VR for medical applications, surgical simulation, visualization, graphics and human-computer interaction.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.