

# DRG-Keyboard: Enabling Subtle Gesture Typing on the Fingertip with Dual IMU Rings

CHEN LIANG, CHI HSIA, CHUN YU\*, YUKANG YAN, YUNTAO WANG, and YUANCHUN SHI, Department of Computer Science and Technology, Tsinghua University; Key Laboratory of Pervasive Computing, Ministry of Education, China

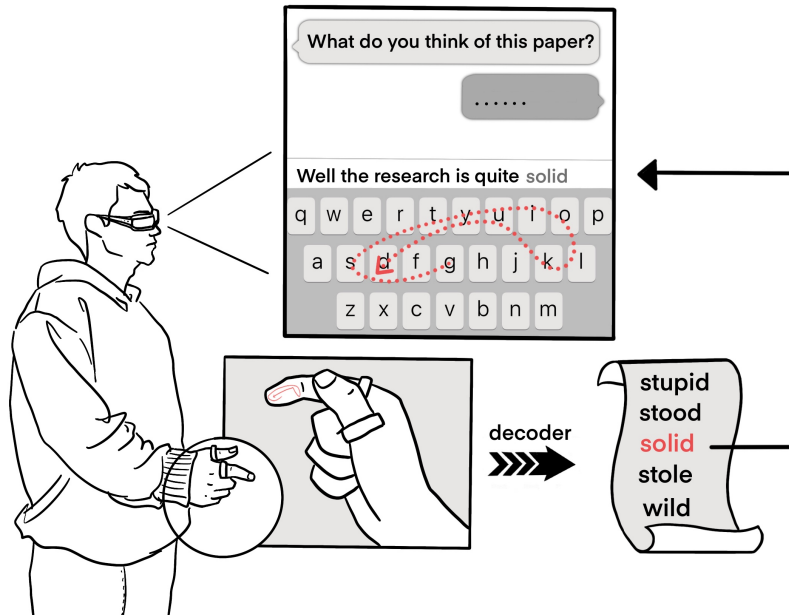


Fig. 1. A typical AR interaction scenario enabled by DRG-Keyboard. The user is chatting with their friend in public, with the chatting window displayed on the AR glasses. The user swipes their thumb on the index fingertip to perform a subtle word gesture, which is decoded by DRG-Keyboard into the desired word.

We present DRG-Keyboard, a gesture keyboard enabled by dual IMU rings, allowing the user to swipe the thumb on the index fingertip to perform word gesture typing as if typing on a miniature QWERTY keyboard. With dual IMUs attached to the user's thumb and index finger, DRG-Keyboard can 1) measure the relative attitude while mapping it to the 2D fingertip coordinates

\*indicates the corresponding author.

Authors' address: [Chen Liang](mailto:liang-c19@mails.tsinghua.edu.cn), liang-c19@mails.tsinghua.edu.cn; [Chi Hsia](mailto:xq22@mails.tsinghua.edu.cn), xq22@mails.tsinghua.edu.cn; [Chun Yu](mailto:chunyu@tsinghua.edu.cn), chunyu@tsinghua.edu.cn; [Yukang Yan](mailto:yukangy@andrew.cmu.edu), yukangy@andrew.cmu.edu; [Yuntao Wang](mailto:yuntaowang@tsinghua.edu.cn), yuntaowang@tsinghua.edu.cn; [Yuanchun Shi](mailto:shiyu@tsinghua.edu.cn), shiyu@tsinghua.edu.cn, Department of Computer Science and Technology, Tsinghua University; Key Laboratory of Pervasive Computing, Ministry of Education, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2474-9567/2022/12-ART170 \$15.00

<https://doi.org/10.1145/3569463>

and 2) detect the thumb's touch-down and touch-up events combining the relative attitude data and the synchronous frequency domain data, based on which a fingertip gesture keyboard can be implemented. To understand users typing behavior on the index fingertip with DRG-Keyboard, we collected and analyzed user data in two typing manners. Based on the statistics of the gesture data, we enhanced the elastic matching algorithm with rigid pruning and distance measurement transform. The user study showed DRG-Keyboard achieved an input speed of 12.9 WPM (68.3% of their gesture typing speed on the smartphone) for all participants. The appending study also demonstrated the superiority of DRG-Keyboard for better form factors and wider usage scenarios. To sum up, DRG-Keyboard not only achieves good text entry speed merely on a tiny fingertip input surface, but is also well accepted by the participants for the input subtleness, accuracy, good haptic feedback, and availability.

CCS Concepts: • **Human-centered computing** → **Text input**.

Additional Key Words and Phrases: text entry, gesture keyboard, fingertip interaction, smart ring

#### ACM Reference Format:

Chen Liang, Chi Hsia, Chun Yu, Yukang Yan, Yuntao Wang, and Yuanchun Shi. 2022. DRG-Keyboard: Enabling Subtle Gesture Typing on the Fingertip with Dual IMU Rings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 170 (December 2022), 30 pages. <https://doi.org/10.1145/3569463>

## 1 INTRODUCTION

With the rapid development of wearable devices and input technology, users are requiring always-available input devices in compact form for convenient input everywhere. Thumb-to-index-fingertip gestures offer new potentials for subtle and precise interactions [50, 52], enabling users to input in spatially or physically restricted scenarios. In such an input modality, the user touches and swipes on the index finger pulp with their thumb tip, as if using a touchpad, to perform input.

TipText [50] and BiTipText [49] have implemented thumb-to-index-tip touch-based text entry with capacitive finger sleeves, proving the feasibility of subtle thumb-tip typing. The work also demonstrates that human's capability to touch a precise absolute position on the index finger pulp is limited, urging TipText to adopt a keyboard-gridding (e.g., dividing the keyboard into a 2x3 grid) strategy for better input recognition.

As an extension of touch-based thumb-tip typing, we envision word gesture typing on the fingertip, where the user imagines a miniature QWERTY keyboard placed on the pulp of the index finger and performs word gestures with the thumb, as illustrated in Figure 1. We propose DRG-Keyboard (abbreviation of Dual Ring Gesture Keyboard), a novel text entry technique using two IMU (inertial measurement unit) rings to enable thumb-to-index-fingertip micro gesture typing with perfect self-bodied haptic feedback.

DRG-Keyboard is an application instantiation of DualRing [29] with optimized prototype design, featuring two wireless IMUs attached to the proximal phalanx of the user's thumb and index finger while removing all the connections and external circuits [29] for better portability. The two IMU rings capture the attitudes and movements of the thumb and the index finger and are combined to acquire their relative attitude and inertial features. We followed DualRing's [29] parameterized approximation method to map the relative attitude into the 2D coordinates on the fingertip. Different from DualRing, we investigated IMU-based touch event detection for DRG-Keyboard, combining the accelerometers' frequency domain data of the two IMUs with their attitude data to recognize the touch-down and touch-up events. The proposed touch event detection algorithm achieved an F-1 score of 98.0% for touch-down and 98.5% for touch-up events. We further analyze the quality of the mapped 2D fingertip coordinates of different patterns and in different environmental interference settings, which was the essential and complementary result to validate the computational feasibility and applicability of DualRing's 2D coordinate mapping algorithm.

To understand users' thumb-to-index-fingertip gesture typing behavior with DRG-Keyboard, we conducted a study to collect users' typing data in two mental models: visual-dominated model and haptic-dominated model. We analyzed the touch point distribution as well as the gesture speed and the subjective perception regarding the two settings. We optimized the general elastic gesture decoder by incorporating a rigid pruning algorithm, a

unigram/bigram language model, and a global distance transform based on the touch point statistics. A thrilling finding is that, a user can perform gesture typing on the fingertip in a natural manner relying less on the visual feedback to get a good result, demonstrating the superiority of DRG-Keyboard in usability.

We further conducted a user study to evaluate the performance and usability of DRG-Keyboard, along with an appending study to compare DRG-Keyboard with the up-to-date Apple Watch Swipe Keyboard in terms of form factor and usage scenario. Results found that all participants achieve an average input speed of 12.9 WPM (68.3% of their gesture typing speed on the smartphone). In the appending study, participants achieved approximately 80% of Apple Watch Swipe Keyboard's input speed with DRG-Keyboard. They also showed their favor of DRG-Keyboard for its one-handedness, less visual attention, privacy preserving, and comfort.

To sum up, our contributions are three-fold:

- We present the concept of DRG-Keyboard, a miniature gesture keyboard on the fingertips, allowing users to input texts by swiping the thumb on the index fingertip with perfect haptic feedback, along with the design considerations and usage process. Our work is also the first to achieve gesture typing on the tiny fingertip with perfect haptic feedback.
- We extend DualRing's [29] prototype to implement DRG-Keyboard, leveraging IMU signals from the wireless rings attached to the thumb and the index finger. The sensing pipeline consists of two key components of mapping finger attitudes to 2D coordinates and recognizing touch-down and touch-up events. We conducted a comprehensive evaluation on touch event detection, 2D fingertip coordinates accuracy, and environmental effects on coordinates mapping to validate the computational feasibility of DRG-Keyboard's sensing schemes. Our results provide complementary insights to DualRing regarding the sensing capability.
- We conducted a comprehensive study of users' fingertip gesture typing behavior, bringing insights to optimize the general elastic decoder for DRG-Keyboard. With such optimizations, DRG-Keyboard can achieve an input speed of 12.9 WPM and is well accepted by the participants for input subtleness, accuracy, good haptic feedback, and availability. Our extended usability evaluation with Apple Watch Swipe Keyboard also showed the advantages of DRG-Keyboard in the form factor to enable unrestricted and always-available text input for heterogeneous displays (e.g., AR, VR, and IoT).

## 2 RELATED WORK

Our work relates to ring-based hand/finger gesture sensing, subtle text entry, and gesture typing. Below we present the related work.

### 2.1 Ring-Based Hand/Finger Gesture Sensing

Smart rings are gaining popularity for their portability and compactness. Taking advantage of the wearing position and sensor potential, a ring-form device plays an important role in hand and finger gesture sensing. Gestures that can be sensed by a ring, or a sensor node attached to the finger, generally involve movement of specific fingers [17, 23] or the entire hand [8, 29, 62], hand postures featuring relative finger states [17, 29], touch gestures where finger-to-finger or finger-to-surface collision occurs [29, 41], the interaction with the ring surface (such as rotating) [15, 44], etc.

To facilitate efficient hand/finger sensing, previous work investigated the capability of various sensors, such as an IR proximity sensor [23], an IMU (inertial measurement unit) [17, 29, 30, 41], an electrical or electromagnetic sensor [11, 48], and an RGB [8] or thermal [62] camera. These sensors can be integrated organically on a ring to achieve the aforementioned sensing purposes. A typical category is camera-based sensing, meaning attaching a camera on the ring to capture pixel-wise hand information. For example, CyclopsRing [8] attached a fish-eye camera on the ring to capture finger movement and hand gesture. ThermalRing [62] used a low-resolution thermal camera to recognizing drawing gestures and thermal tags. Although camera-base solutions can sense

precise pixel-wise information, they suffer from power consumption and computation complexity, and may fail in detecting subtle movements[29]. Another category is IMU-based sensing, leveraging the IMU's advantage in detecting orientations and micro-vibrations. For example, LightRing [23] combined an infrared proximity sensor with a 1-axis gyroscope to achieve 2D cursor-based input on any surface. The proximity sensor was used for measuring finger flexion while the gyroscope was used for measuring finger rotation. ThumbRing [45] enabled thumb-to-palm interaction as if using a pointer, which is achieved by attaching 2 IMUs, one to the user's thumb and one fixed to the user's wrist as the reference pose.

As most related to our work, DualRing [29] combined dual IMU rings with an external high-frequency AC circuit. The two IMUs were used for mapping relative attitudes to fingertip coordinates as well as recognizing complex gestures while the external circuit was used for precise pinch detection. Such a prototype provides strong capability in full-hand gesture sensing, though the redundant wires and external circuits prevent it from being lightweight and portable. Given previous work where inertial-based finger contact and touch detection [18, 41] was widely researched, we aim to develop an IMU-based solution for accurate thumb-to-finger touch event detection that facilitate more compact and efficient hardware form for DRG-Keyboard in this work.

## 2.2 Thumb-to-finger Interaction

Touching the thumb to the fingers is a privacy-preserved subtle interaction while keeping naturalness and expressivity, and has been widely explored by researchers. Such interactions could be sensed through various techniques, including camera[8, 31, 42], magnetic sensors[3, 9, 10], capacity sensors[44], EMG[40], microphone together with a gyroscope[61] and so on. Heads[42], necks[31], wrists[14], proximal finger segments[3, 8], fingertips[49, 50] and fingernails[9] are the typical positions where the sensors are attached to.

Previous work has investigated the design of thumb-to-finger gesture space, typically from the dimensions of contact position[21, 42] and event type (e.g., discrete tapping[14, 31] v.s. continuous control[42]). For example, FingerInput[42] thoroughly discussed the gesture space of thumb-to-finger microgestures and implemented a camera-based system which could accurately detect touch points and finger flexion, thus empowering a full extensibility to a broader set of gestures. DualRing[29] sensed not only the absolute hand gesture relative to the ground but also the relative pose and movement among hand segments with two IMU rings worn on thumb and index finger. Among the various microgestures, tapping, sliding and gesture input were most commonly used and researched. Tapping with thumb is an intuitive discrete action as if pressing on a virtual button. For example, PinchWatch[31] programmed tapping on different fingertips to trigger different functionalities, while in DigiTouch[47], different finger segments could be mapped to different characters, in turn supporting typing. The slider is a common application leveraging continuous control, where user swipes along the finger or around the finger as 1D input[3, 46]. On the other hand, 2D input takes the form of performing gestures (e.g. digits[61], characters[21]) without visual attention. DRG-Keyboard is the first to explore gesture typing on fingertip. Different from gesture input where the trace drawn by the user is to be categorized into several classes, gesture typing requires a better trace quality and a powerful decoder.

## 2.3 Subtle Text Entry

The development of user interfaces and wearable devices cast the demand for input technologies in different scenarios. Among those scenarios, subtle text entry is most challenging and has been widely researched. The difficulty is that subtle gestures as input often have lower SNR [55], thus requiring a well-designed modality or an advanced decoder. The subtleness in text entry consists of two intrinsic dimensions: the subtleness of the input interface (e.g., a smartwatch touchscreen [2, 37], a touchable finger glove [49, 50], and a ring-size touch surface [24]) and the subtleness of the input behavior (e.g., a micro thumb-to finger gesture [49, 50], the hand orientation [20], the tilt [54], and the finger pressure [63]).

As a typical scenario of subtle text input, typing on a tiny surface has been widely researched in previous work. Yi et al. [55, 56] investigated the touch point distribution and showed the feasibility to apply the ordinary QWERTY keyboard on a smartwatch touchscreen. SwiperRing [37] redesigned the keyboard layout, fitting the keys to the bezel of the dial to adapt gesture typing on a smartwatch by enhancing the angular distinguishability of the traces. ThumbText[24] organized the keys into a grid layout to enable typing on a ring surface with hybridization of touch and swipe gestures. Typing with micro gestures is another major type of subtle text entry, playing an important role in scenarios with physical constraints or requiring high privacy. TipText [50] and BiTipText [49] allowed users to type subtly on an ordinary or a split QWERTY keyboard on the index fingertip with capacitance finger sleeves. Force Board [63] leverage implicit finger pressure collected from a smartphone, mapping it to a sliding window for key selection to enable subtle text input.

## 2.4 Gesture Typing

Gesture typing, an elegant input technique proposed by Zhai and Kristensson [25, 58, 59], has been widely adopted and deployed on various commodity devices, such as iPhones [60] and Android phones. Recent work has extensively explored gesture typing in different forms and modalities. The exploration includes adaption in devices of different physical forms, such as HMDs [57], tablets [6], smart watches [53], and rings [20]. Researchers also delved deep into the input modalities and mental models that benefited from gesture input. For example, Bi et al. [6] proposed a bimanual gesture keyboard allowing a user to input gestures with both hands simultaneously, boosting the efficiency on large-screen devices. Zhu et al. [64] proposed a novel decoding algorithm to improve the decoding accuracy of eyes-free gesture typing, while Yang et al. [51] proposed starting gesture typing from a fixed key to improve the usability in eyes-free scenarios. RotoSwipe [20] and Vulture [34] used IMUs and optical trackers respectively to capture users' in-air hand swiping strokes for gesture input. Yu et al. [57] investigated using head rotation to perform gesture typing, while Kumar proposed a gesturing typing method hybridizing gaze swipe with touch input [27]. Yeo et al. [54] developed a novel method leveraging the tilt angle of the smartphone for gesture typing.

As an inspiration of previous work, our work aims to enable gesture typing on the fingertip with self-bodies haptic feedback, which has not been explored and is a challenging problem due to the subtleness of finger movement and the constraint that the sensing solution should not impede self-bodies haptic feedback.

## 3 DRG-KEYBOARD: DESIGN

In this section, we introduce the input modality, the usage process, and the design considerations of DRG-Keyboard. We first introduce the mental model and basic modality of DRG-Keyboard, followed by the design questions. We explain how we justify the details in consideration of our design rationales to achieve a good fingertip gesture keyboard design. Finally, we present the finalized usage process along with the UI design.

### 3.1 Gesture Typing on the Index Fingertip

The basic mental model and input method of DRG-Keyboard is that, an imaginary QWERTY keyboard is placed on the pulp of the index finger's distant segment and the user uses the thumb as a pointer to swipe on the index fingertip to draw traces. A few design details, including the imaginary keyboard's boundary and scale, the word gesture being absolute or relative, and the word confirmation mechanism are to be determined before the design is finalized. We focus on answering the following three design questions (DQ) to clarify the necessary details.

- **DQ1: What is the proper boundary and scale of the imaginary keyboard?** A pilot interview shows there are three types of probable QWERTY keyboard placement on the index finger: 1) the keyboard covers the distant and the middle segments and is split by the distant knuckle; 2) the keyboard is placed on the distant segment, covering the whole pulp area of it; and 3) the keyboard is placed on the distant

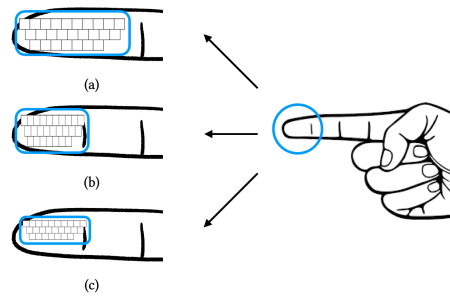


Fig. 2. Probable QWERTY keyboard placement on the index finger. (a) The keyboard crosses the distant knuckle. (b) The keyboard covers the whole pulp area. (c) The keyboard approximately covers the top  $\frac{2}{3}$  pulp area.

segment, approximately covering the top  $\frac{2}{3}$  pulp area of it, as shown in Figure 2. For 1), the distant knuckle may impede the smooth movement of the thumb, while the bending of the two segments may lead to inconsistency between the user's mental model and the real touch. Further, swiping the thumb on two segments is not as subtle as we expected. For 2), fitting a QWERTY keyboard to the whole pulp area would lead to a vertical stretching transform of the keyboard, which is proved inferior for indirect touch input [51]. Besides, touching the lower boundary with the thumb in such a setting is not comfortable. For the considerations above, we choose 3) as the imaginary keyboard boundary.

- **DQ2: Should the position of the imaginary keyboard be absolute or relative to the first touching point?** In our work, we adopt a relative keyboard strategy, meaning the trace starting from a fixed key on the keyboard. Following [51], we choose key "G" as the start point of a trace. The reasons why we choose to use the relative strategy includes: 1) traditional gesture keyboard decoding algorithms (e.g., SHARK2 [25]) is sensitive of the trace's start point and absolute position; and 2) TipText [50] shows the human's capability to reach a precise key on a fingertip QWERTY keyboard is limited, prompting the design of coarse-grained keyboard layout (e.g., a  $2(\text{vertical}) \times 3(\text{horizontal})$  grid) to alleviate high-resolution absolute touch input.
- **DQ3: How to accomplish word confirmation after inputting a gesture?** Possible confirmation strategies for touch-based keyboard mainly include: 1) defining special gestures to trigger and accomplish the confirmation (e.g., wrist rotation to select candidates, fast swipe or double tap to trigger word selection phase, etc.); and 2) dividing the input process into two phases - the input phase and the selection phase. In our design, we adopt the second strategy because alternatively performing swiping and selection gestures may lead to frequent switches in the user's behavior model, fragmenting the coherent input process. Moreover, since the input phase covers the most natural gesture space, a distinguishable gesture for word selection would require additional physical demand. It is worth emphasizing that, the two phases in our design share the same swiping gesture space - in the selection phase, the user swipe from a fixed point in a specific direction to select the desired word. Thus our 2-step input strategy alleviates frequent mode switch while not being necessary to design physical-demanding gestures.

### 3.2 Usage Process and UI Design

The usage of DRG-Keyboard is that the thumb, working as a pointer, draws a trace on an imaginary QWERTY keyboard on the index fingertip. A user performs gesture typing to input a word on the fingertip in the following five steps:

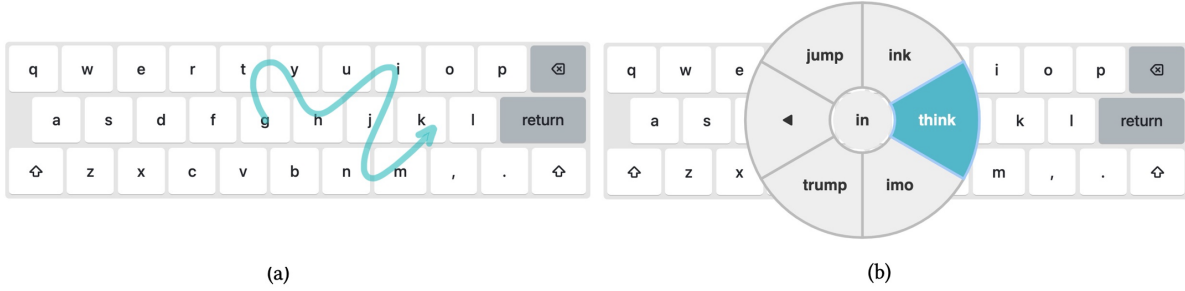


Fig. 3. Keyboard layout and UI design: (a) The full QWERTY keyboard layout with trace feedback for gesture input. (b) The wheel navigation widget for candidate selection.

- Touch down the thumb to begin the input of a word.
- Draw the pattern of the desired word with the thumb starting from a fixed key - "G".
- Touch up the thumb to trigger the candidate selection phase.
- Touch down the thumb and swipe in the corresponding direction (or keep still) to select the desired word or trigger cancel. The top-1 word is always at the center and swiping right always indicates cancel.
- Touch up the thumb to confirm the selection.

Users delete a word by performing a tap (touching down then immediately touching up) with the thumb at the gesture input phrase.

The UI design of DRG-Keyboard is shown in Figure 3. In the gesture input phase, the UI features a full QWERTY keyboard with real-time stroke feedback, as shown in Figure 3 (a). In the candidate selection phase, a wheel navigation widget containing the top 6 candidates along with a cancel button is displayed, as shown in Figure 3 (b). The selected word is highlighted.

#### 4 DRG-KEYBOARD: TECHNICAL DETAILS AND IMPLEMENTATION

In this section, we illustrate the technical details - including the hardware setup and the sensing algorithms - of DRG-Keyboard. We explain how to 1) calculate the relative attitude while mapping the relative attitude to the 2D fingertip coordinates and 2) detect thumb-to-index-fingertip touch-down and touch-up events based on the combination of frequency-domain features and geometric features of dual IMUs.

##### 4.1 Mapping Relative Attitudes into 2D Fingertip Coordinates

We follow DualRing's [29] coordinate mapping model to build our 2D fingertip coordinate mapping algorithm. According to Liang et al.'s analysis, such a fingertip coordinate mapping task can be approximated as projecting a pointer vector onto a target plane. Let  $M_R$  be the relative transform matrix between the two IMUs and  $\mathbf{m}_R$  is the flatten 1-dimensional vector of  $M_R$ , the coordinates  $x, y$  on the target plane can be represented in linear combinations of the elements in  $\mathbf{m}_R$ :

$$x = \mathbf{p} \cdot \mathbf{m}_R, y = \mathbf{q} \cdot \mathbf{m}_R \quad (1)$$

, where  $\mathbf{p}, \mathbf{q}$  are the fitting vectors to be solved, each containing 9 parameters.

To acquire the undetermined parameters  $\mathbf{p}, \mathbf{q}$ , the user is required to sample multiple data points for parameter fitting. During fitting, the user draws multiple horizontal and vertical unit vectors on the fingertip based on their perception to sample "horizontal pairs" and "vertical pairs". We denote  $S_h = \{(\mathbf{m}_i, \mathbf{n}_i), i = 1, 2, \dots, k\}$  the set of horizontal sampled pairs and  $S_v$  the set of vertical sampled pairs. A sampled pair  $(\mathbf{m}_i, \mathbf{n}_i)$  is the flattened vectors

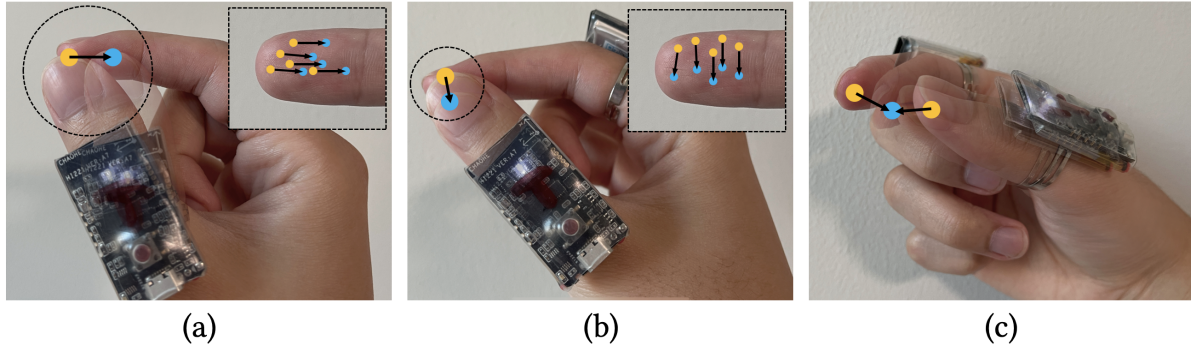


Fig. 4. Illustration of the calibration process: (a) horizontal calibration, (b) vertical calibration, and (c) calibration of touch attitudes.

of the relative transform matrix  $m_R$  of the start point and the end point of the corresponding traces (the start point and the end point of a horizontal (vertical) pair form a horizontal (vertical) base vector  $(1, 0)^T ((0, 1)^T)$ ). An least square optimization with regulation term is adopted to learn the relative movement vector  $(\Delta x, \Delta y)$ . We aim to fit the relative movement for its better perception by the user rather than the absolute position on the fingertip. We rewrite DualRing's [29] objective loss function for optimization in a more compact form:

$$\begin{aligned}
 L_p &= \sum_{(m,n) \in S_h} \|\mathbf{p}(m - n) - 1\|^2 + \sum_{(m,n) \in S_v} \|\mathbf{p}(m - n) - 0\|^2 + \lambda \|\mathbf{p}\|^2 \\
 L_q &= \sum_{(m,n) \in S_h} \|\mathbf{q}(m - n) - 0\|^2 + \sum_{(m,n) \in S_v} \|\mathbf{q}(m - n) - 1\|^2 + \lambda \|\mathbf{q}\|^2
 \end{aligned} \tag{2}$$

, where  $\lambda \|\mathbf{p}\|^2$  and  $\lambda \|\mathbf{q}\|^2$  ( $\lambda = 0.001$ ) are regulation terms to prevent overfitting. By minimizing the loss function  $L_p$  and  $L_q$  respectively, we acquire the optimized parameter  $\mathbf{p}$  and  $\mathbf{q}$ . After acquiring  $\mathbf{p}$  and  $\mathbf{q}$ , the relative movement in  $x$ - and  $y$ - axes can be represented as:

$$\begin{aligned}
 \Delta x &= s_x \cdot \mathbf{p} \cdot \Delta m_R \\
 \Delta y &= s_y \cdot \mathbf{q} \cdot \Delta m_R
 \end{aligned} \tag{3}$$

, where  $\Delta m_R$  is the difference of  $m_R$  between two consecutive frames and  $(s_x, s_y)$  is the scaling parameters influencing the sensitivity on  $x$ - and  $y$ -axes, which can be dynamically adjusted by the user depending on the desired applications.

Figure 4 (a) and (b) show the detailed calibration process of the vertical and horizontal fitting vectors  $\mathbf{p}$  and  $\mathbf{q}$  from the user's perspective. Taking (a) as an example, when calibrating the horizontal vector, the user should pose the thumb on the index fingertip at the start position (the yellow dot) and pressed a key to begin recording. Then they moves the thumb to the end position (the blue dot) to draw a unit horizontal vector and press a key to end recording. For each calibration, the user should repeat the above process five times to collect five traces (or five horizontal sample pairs). Notably, since the algorithm aims to fit relative movement instead of absolute position, the user is encouraged to collect the traces at different positions (e.g., Figure 4 top right) to improve data diversity. Similarly, the user draws and records vertical traces as shown in Figure 4 (b).



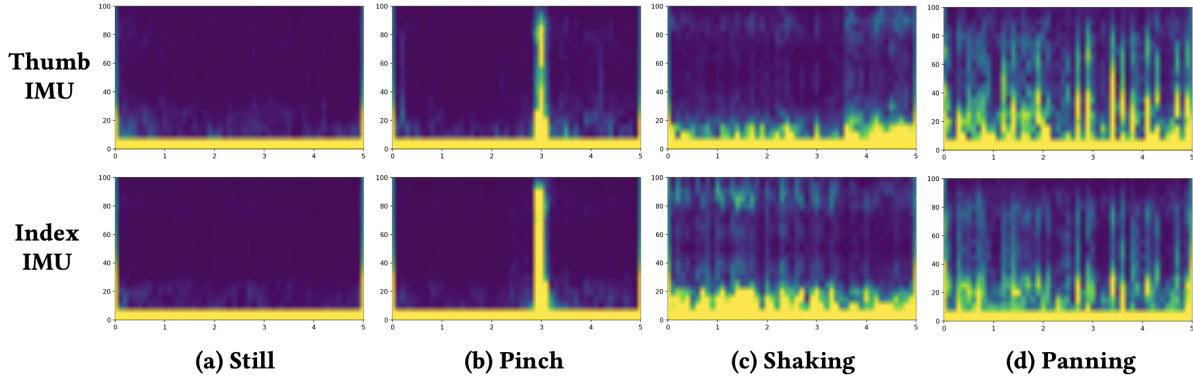


Fig. 5. The STFT maps of different thumb-to-index-finger gestures: a) keeping still; b) pinching; c) shaking; and d) panning.

#### 4.2 Detecting Touch-Down and Touch-Up Events

We incorporate 1) the variation of the relative attitude and 2) the synchronous frequency-domain feature of the accelerometers to recognize the touch-down and touch-up events.

We follow the same parameterized method as the previous subsection to fit a parameter  $\mathbf{r}$  such that  $\mathbf{r}\mathbf{m}$ , where  $\mathbf{m}$  is the 1-D flattened vector of the relative attitude matrix, is sensitive to the relative finger movement in the touching direction. Assuming  $S_t$  is the set of sampled pairs in the touching direction (for  $(\mathbf{m}_i, \mathbf{n}_i) \in S_t$ ,  $\mathbf{m}_i$  referring to the vector in touch-up state and  $\mathbf{n}_i$  referring to the vector in touch-down state) and  $S_h$  and  $S_v$  are the horizontal and vertical sample sets mentioned above, the objective function for optimization is:

$$L_r = \sum_{(\mathbf{m}, \mathbf{n}) \in S_h} \|\mathbf{r}(\mathbf{m} - \mathbf{n}) - 0\|^2 + \sum_{(\mathbf{m}, \mathbf{n}) \in S_v} \|\mathbf{r}(\mathbf{m} - \mathbf{n}) - 0\|^2 + \sum_{(\mathbf{m}, \mathbf{n}) \in S_t} \|\mathbf{r}(\mathbf{m} - \mathbf{n}) - 1\|^2 + \lambda \|\mathbf{r}\|^2 \quad (4)$$

By optimizing  $L_r$ , we can obtain an indicator function  $i_{att}$  in the touching direction – For the last consecutive  $N$  ( $N = 10$ ) frames of relative attitude vector  $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_N$ , we have:

$$i_{att} = \sum_{i=1}^{N-1} \mathbf{r}(\mathbf{m}_{i+1} - \mathbf{m}_i) \quad (5)$$

Similar to the calibration of horizontal and vertical vectors, the user is required to collect touch-up (the yellow dots) and touch-down (the blue dot) attitudes as shown in Figure 4 (c). The user is also encouraged to collect touch-up and touch-down attitudes at different hand postures.

As for the frequency-domain features, previous literature [28] shows that accelerometers attached to the hand can capture bio-acoustic signals indicating within-hand gestures. To recognize the thumb-to-index-tip gestures used in DRG-Keyboard, We apply a Short Time Fourier Transform (STFT, Hann window, 200 FPS, segment length 40) on both the thumb IMU and the index IMU. We fetched the latest 1000 frames of each accelerometer's data and compute the STFT maps on data from each accelerometer axis, producing three feature maps  $X$ ,  $Y$ , and  $Z$ . Then we combined the three feature maps by taking the max value at each position, producing a feature map  $F$  of the same shape.

As shown in Figure 5, we can observe that different thumb-to-index gestures yield different STFT feature maps. Typically, a thumb-to-index-finger touch-down (pinch) results in evenly distributed signal strengths at different frequencies, the figure of which looks like a "vertical bright line". Based on this finding, we design a simple but

effective method to detect touch-down events with a pre-defined vertical edge filter  $F$ . Assuming the STFT maps of the last 200 ms are  $S_T$  and  $S_I$ , we first remove the frequency lower than 10 Hz, acquiring  $S'_T$  and  $S'_I$ . Then we calculate the inner products  $F \cdot S'_T$  and  $F \cdot S'_I$ , where  $F(h \times v)$  is the edge filter such that  $F[\frac{h}{4}] = F[\frac{3h}{4}] = -1$  and  $F[\frac{h}{4} : \frac{3h}{4}] = 1$ . We represent the frequency indicators  $i_{frq-T}$  and  $i_{frq-I}$  as:

$$i_{frq-T} = \frac{F \cdot S'_T}{(1 + \text{mean}(S'_T))}, i_{frq-I} = \frac{F \cdot S'_I}{(1 + \text{mean}(S'_I))} \quad (6)$$

Finally, the touch-down signal  $S_{down}$  and the touch-up signal  $S_{up}$  can be represented as:

$$\begin{aligned} S_{down} &= i_{frq-T} > \text{thred}_1 \wedge i_{frq-I} > \text{thred}_1 \wedge i_{att} > \text{thred}_2 \\ S_{up} &= i_{att} < -\text{thred}_3 \end{aligned} \quad (7)$$

, where  $\text{thred}_*$  are user-related threshold parameters that can be adjusted in the calibration phase.

### 4.3 Prototypical Implementation

We prototype DRG-Keyboard with two customized 9-axis wireless IMU module and a customized receiver module. Each IMU module consists a BMI-055 6-axis IMU and a QMC5883L 3-axis magnetic sensor. The 9-axis sensor data (acceleration, gyroscope data, and geomagnetic data) are streamed to and processed by an NRF52832 Bluetooth Low Energy (BLE) SoC, on which an extended Kalman filter (EKF) is implemented to compute the module's attitude relative to the ground (or AHRS, represented as 3-axis Euler angle). The 9-axis sensor data and the 3-axis attitude data are transmitted to a PC via an RFX2401C RF front-end integrated circuit chip asynchronously at 200Hz. The whole IMU module is powered by a 300mAh Lithium battery with 8 hours of (working) battery life. The hardware form and the wearing example were shown in Figure 6. The customized receiver module receives the 9-axis sensor data and the 3-axis attitude data from the two IMU modules with an RFX2401C chip. The data is first processed by an NRF52832 SoC and then stream to a PC using USB serial protocol via a CH340E chip (460800 Baud Rate).

Since DRG-Keyboard did not have special requirements on the sensor specifications (e.g., the BMI-055 and the QMC5883L were common commodity sensors), we thought it would be feasible to prototype the wireless IMU nodes on standard off-the-shelf hardware (e.g. an Arduino NANO 33 Sense with an LSM9DS1 9-axis IMU module and an NRF52840 SoC) with little effort. However, we chose the customized modules in our implementation for optimized form in size (e.g., we found the Arduino NANO 33 board was too long to impede the thumb from bending upward, while our module was approximately  $\frac{3}{4}$  in length and could alleviate this problem), power management (e.g., using an external battery to power the Arduino board would lead to additional volume and weight), and multi-node data transmission (optimized in our module).

As for the PC side, We implement the sensing algorithms using Numpy and Scipy in Python on a 13 inch MacBook Pro. The algorithms run on one 2.3GHz Intel CPU core and the FPS of the whole pipeline including data streaming and algorithm running is 200. The delay of reporting a touch event is approximately 100ms due to the use of spectrum features. The computational cost of the algorithms mainly comes from matrix multiplication (Equation 3, 4) and STFT map computation, which has been proved computational-friendly for always-on deployment on mobile devices (e.g., smartphone and smartwatch [36]). Thus we assure that the current implementation of DRG-Keyboard is in an efficient computational form that is capable for deployment on mobile devices like AR and VR glasses.

### 4.4 Evaluation

To validate our algorithms' computational feasibility, we conducted three experiments to evaluate 1) the performance of the touch detection algorithm, 2) the performance of the coordinates mapping algorithm, and 3) the

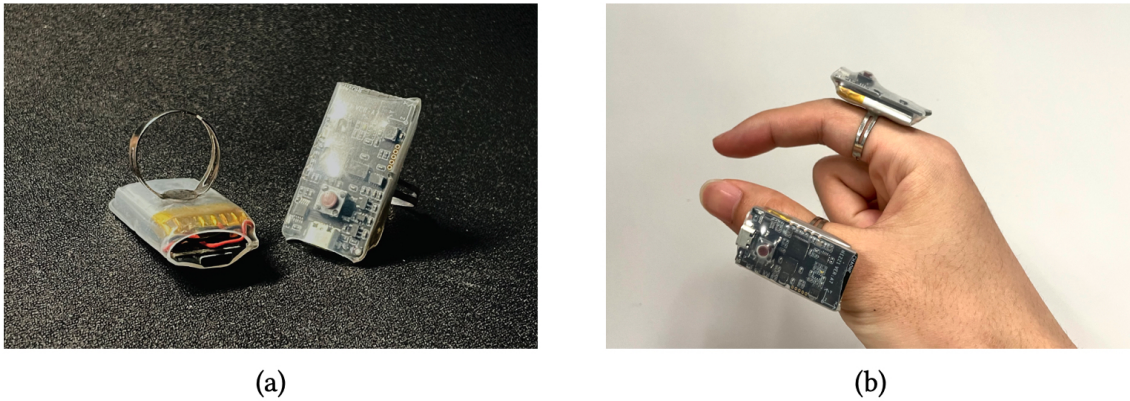


Fig. 6. DRG-Keyboard hardware prototype. (a) Two IMUs are attached to the rings to capture fingers' attitudes; (b) a wearing example.

Table 1. The precision, recall, and F-1 score of touch detection (with standard deviation).

Gesture	Precision	Recall	F-1 Score
Touch-down	99.0 (1.3) %	97.0 (1.6) %	98.0 (1.0) %
Touch-up	99.4 (1.5) %	97.6 (2.6) %	98.5 (1.4) %

effects of different environments on coordinates mapping. The apparatus settings for the three experiments were same as described in Section 4.3.

**4.4.1 Touch Detection.** For touch detection, we recruited 10 users (2 females) to participate in a real-time evaluation of touch-down and touch-up detection accuracy. After calibrating with the algorithms described in Section 4.2 (Equation 4,  $|S_t| = |S_h| = |S_v| = 5$ ), the users were required to adjust the threshold in Equation 9 freely (with real-time detection result feedback) until they found themselves behaved comfortably and performed well, where the system was sensitive to touch events and rarely yielded false positives. Then they performed touch-downs and touch-ups 100 times for each (TP+FN=100). The count of false positives (FP) was recorded during the whole trail. We measured the precision, recall, and F-1 score. Results in Table 1 showed our touch detection algorithms generally had a good performance detecting both touch-up (F-1: 98.5%) and touch-down (F-1: 98.0%) events. We also found users tended to choose a threshold of higher precision (e.g., 99.0% for touch-down and 99.4% for touch-up) than recall, meaning they were less tolerant of unintentional mis-touches.

**4.4.2 2D Coordinates Mapping.** We conducted a calibration experiment followed by a pattern drawing experiment to evaluate our 2d coordinate mapping algorithm and grasp an understanding of how well users perform fingertip gestures using our technique. We recruited 10 users (5 females) from the campus to participate in the study.

For the calibration evaluation, users were first guided through the calibration process described in Section 4.1, sampling 5 horizontal left-to-right traces and 5 vertical top-to-bottom traces for fitting (as described in Section 4.2) and additional 10 inverse traces (5 right-to-left and 5 bottom-to-top) for evaluation. We applied linear regression to each trace to measure the angular bias and computed the standard deviation on y coordinates for horizontal traces and x coordinates for vertical traces. Though it's intuitive and common to measure goodness

of fit with R-square, a linear model performs no better than reporting a constant mean value when the trace is nearly horizontal or vertical, so we use SD as a measurement for straightness. The average delta angle was  $6.09^\circ$  (left-to-right,  $SD=4.75^\circ$ ),  $6.33^\circ$  (right-to-left,  $SD=4.91^\circ$ ) for horizontal traces and  $6.92^\circ$  (top-to-bottom,  $SD=5.49^\circ$ ),  $6.48^\circ$  (bottom-to-top,  $SD=6.30^\circ$ ) for vertical traces. The standard deviation values were 0.045 ( $SD=0.029$ ), 0.049 ( $SD=0.024$ ), 0.047 ( $SD=0.023$ ), and 0.043 ( $SD=0.025$ ) for left-to-right, right-to-left, top-to-bottom, and bottom-to-top, respectively. We visualized 10 users' calibration result in Figure 7, from which we observed a good consistency of different traces in both direction and smoothness.

Since the actual patterns in gesture typing might be complicated, we included three patterns - circles, squares, and isosceles triangles - in the study to further figure out the performance of our technique in drawing more complex shapes. After calibration, the three target patterns were displayed on the screen for reference. For each drawing sample, the user pressed a start button, drew the pattern on the fingertip, and pressed a stop button to end recording. During the drawing, no visual feedback on the canvas was given to the user. The only instruction given to the user was they controlled the cursor and drew the desired pattern mainly based on haptic feedback and in a way consistent with their calibration behavior. After drawing a pattern, the drawn trace was displayed on the canvas. As a comparison, the user was required to accomplish the same task with the trackpad of a 16-inch MacBook Pro with no visual feedback. Each user drew each pattern with each technique (ring v.s. trackpad) for 5 times and the order of techniques and patterns were shuffled to alleviate order effects. We collected a total of  $10 \text{ (users)} \times 3 \text{ (patterns)} \times 2 \text{ (techniques)} \times 5 \text{ (times)} = 300$  traces from the study. It was worth mentioning that since both techniques were for relative cursor control, we did not restrict the position and the scale of the target pattern (neither did we fix a default pattern on the fingertip) so that the user could 1) draw the pattern based on their haptic perception in a more natural manner and 2) collect more diverse patterns at different positions and scales on the fingertip. The only instruction we gave to the user was that they should move their fingers the same way as they did in the calibration and draw the pattern with the same shape as the target pattern.

We measured 1) the normalized distance between the start point and the end point of each trace and 2) the shape similarity between the trace and the target shape for the collected traces of the two techniques (ring and trackpad). Assuming the typical size of human fingertips 16–20 mm in diameter [13] and the comfortable operating area being the top  $\frac{2}{3}$  pulp (based on our pilot interview in Section 3.1), we also converted the reported normalized distances for the ring setting into spatial distances as an intuitive reference. For 1), we first normalized them into a unit region since patterns were drawn in arbitrary size. Using our technique, the start-end distance was 0.261 on average ( $SD = 0.183$ ), or approximately 3.1mm, slightly outperformed the metrics from trackpad (0.274,  $SD = 0.154$ ). We attributed it to the haptic self-feedback from drawing on the fingertip. For 2), dynamic time warping (DTW) algorithm[38] was applied to measure the similarity. Both the drawn and target patterns were resampled to 100 points. we calculated the similarity by averaging the distances between mapped points, with the result of 0.161 ( $SD = 0.055$ ), or approximately 1.9mm, for ring and 0.086 ( $SD = 0.037$ ) for trackpad. Similarity varied slightly between different patterns (0.162 ( $SD = 0.066$ , 1.9mm) for circles, 0.170 ( $SD = 0.042$ , 2.0mm) for squares and 0.153 ( $SD = 0.053$ , 1.8mm) for triangles). Figure 8 illustrated some of the good and poor samples. We classified poor samples into three main categories: 1) shape deformation (which might be reasoned from either mapping algorithm or user's drawing inaccuracy), 2) redundant trace caused by user's misbehavior (e.g. relaxing their finger before pressing the stop button), and 3) rotation between the applied and calibrated coordinates.

**4.4.3 Environmental Effects on Coordinates Mapping.** The coordinates mapping algorithm is based on the ideal assumption that the two IMUs always give reliable attitudes that share the same ground coordinates and would not drift or warp over time. For the 9-axis IMUs used in DRG-Keyboard, the correction phase in the Extended Kalman Filter would compensate the Roll angle and the Pitch angle with the global gravity reference and the Yaw angle with the geomagnetic vector, ensuring no accumulative error (or drift) yielded in the Roll, Pitch, and Yaw angles. However, as mentioned by Liang et al. [29], the attitudes of the IMUs could be interfered by the warping

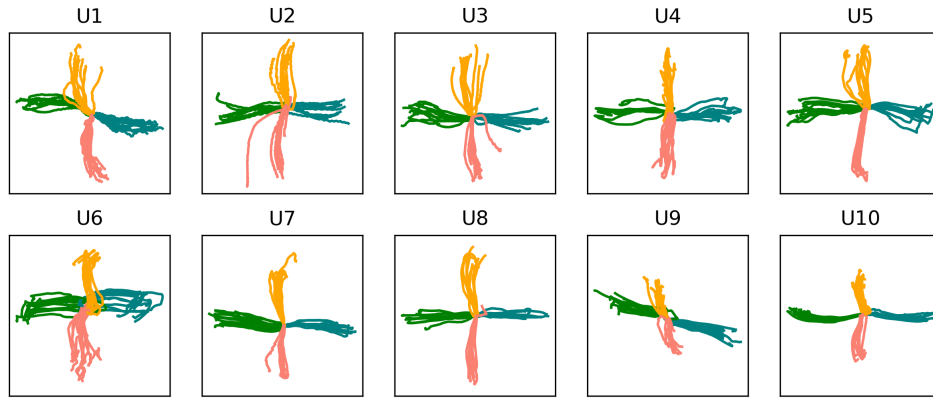


Fig. 7. Visualization of the calibration, consisting of 10 horizontal and 10 vertical traces for each user. Since Equation 1 reports a relative coordinate, we set the start point of each trace to (0,0). Traces in different direction were distinct in color. The x- and y- axes in each figure are in the same scales.

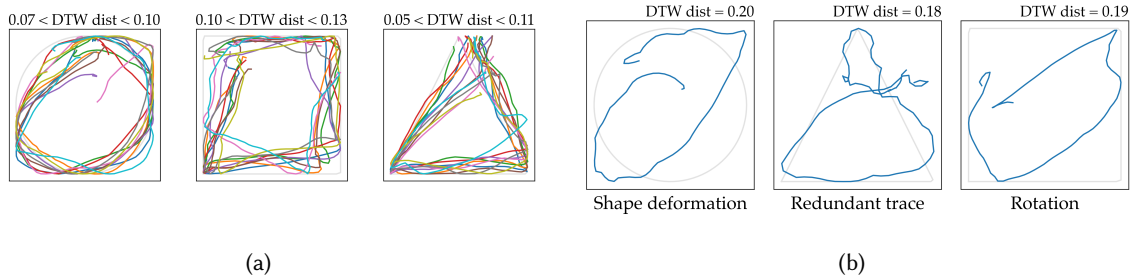


Fig. 8. Pattern examples (colored traces). Target shapes are displayed in grey as a reference. (a) Good cases. (b) Poor cases.

of the magnetic field (especially in indoor scenarios). To this point, an essential research question for practical deployment is how environmental issues (e.g., postures and positions) affect the quality of coordinates mapping.

Therefore, we conducted a cross-environment validation on the coordinates mapping quality to examine the persistency of the mapping algorithm against environment switching (e.g., calibrating in one position and using in another). The study was conducted in a room of approximately  $20m^2$  with 5 electric appliances (1 TV, 1 air conditioner, 1 refrigerator, and 2 laptops) and an open outdoor area. Since magnetic interference is the major noise source of the IMU's attitude, we measured the magnetic field of the room and chose three indoor positions with different magnetic interference levels - at the corridor with no magnetic source nearby (P1,  $50\mu T$ ), near the TV ( $< 30cm$ ) with weak magnetic interference (P2,  $50 + 10\mu T$ ), near the laptop speaker ( $< 10cm$ ) with strong magnetic interference (P3,  $50 + 50\mu T$ ) - and an outdoor position (P4,  $45\mu T$ ) for evaluation. For each position, the user was required to collect the calibration data with different hand orientations and postures to cover the variation brought by posture change. The study collected a total of  $4$  (settings)  $\times$   $5$  (times) =  $20$  calibration records.

Table 2. Results for self-fitting pairs in the cross-environment validation experiment.

Environment	angle bias (°)		straightness		orthogonality
P1	6.77 (SD=5.70)	7.13 (SD=5.63)	0.030 (SD=0.021)	0.035 (SD=0.021)	3.23 (SD=3.44)
P2	5.25 (SD=4.10)	4.93 (SD=4.09)	0.026 (SD=0.016)	0.027 (SD=0.014)	2.43 (SD=1.53)
P3	4.04 (SD=3.07)	5.33 (SD=4.33)	0.023 (SD=0.014)	0.036 (SD=0.015)	1.66 (SD=1.05)
P4	5.77 (SD=3.79)	7.33 (SD=5.42)	0.024 (SD=0.014)	0.031 (SD=0.017)	4.90 (SD=1.60)

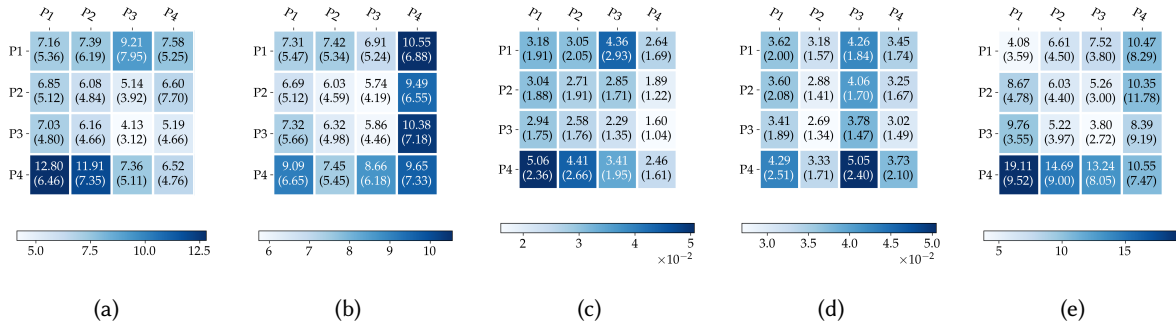


Fig. 9. Evaluation results for environmental effects on coordinate mapping. Different postures and magnetic inference conditions were tested. *ST / SI* stand for *Stand / Sit*, and *R / M / O* stand for *Room / Near magnetic sources / Open area* respectively. (a) angles for horizontal traces, (b) angles for vertical traces, (c) RMSE for horizontal traces, (d) RMSE for vertical traces, (e) Orthogonality.

Each record refers to 5 horizontal left-to-right traces and 5 vertical top-to-bottom traces for fitting (as described in Section 4.2) and additional 10 inverse traces (5 right-to-left and 5 bottom-to-top) for validation.

We conducted the cross evaluation by fitting the calibration parameters  $\mathbf{p}$  and  $\mathbf{q}$  with one calibration record  $A$  and applying the parameters on another record  $B$  to map the traces of record  $B$  into 2D relative coordinates. Such operations generated  $20 \times 20 = 400$  cross-fitting pairs  $(A,B)$  and each pair contains 20 traces. We measured the angle biases, the straightness (measured by standard deviation, consistent with the calibration experiment), and the orthogonality (represented in the included angle) of the horizontal and vertical traces for each pair. The results for self-fitting pairs  $((A,A))$ , fitting and evaluating on the sample record) were listed in Table 2, and the within- and cross-environment results were shown in Figure 9. By comparing self-fitting and within-environment (the diagonal in Figure 9) results, we could observe that changes on hand orientation and postures merely affected the mapping algorithm. Magnetic interference did influence the coordinate mapping quality comparing non-diagonal elements to diagonal, but the effect was acceptable for the tested indoor environments (P1, P2 & P3). Warped magnetic field served as a shifted reference for yaw angles, but since we leveraged the relative attitude of the thumb against index finger, the interference would be counterbalanced as long as the magnetic field held at the same direction on both fingers. These results demonstrated the robustness of our technique under different environment and user postures.

## 5 STUDY 1: UNDERSTANDING THUMB-TO-INDEX-FINGERTIP GESTURE TYPING BEHAVIOR

After determining the sensing schemes and the basic modality design, we conducted a user study to investigate users' thumb-to-index-fingertip gesture typing behavior. We were interested in users' input behavior in two

mental models - visual-dominated input and haptic-dominated input, both of which are with visual feedback. Results from this study could ground insights of users' gesture typing behavior on the fingertip, which has not been researched before.

### 5.1 Participants and Apparatus

We recruited 10 participants (2 females, all ESL) from the local campus. The average age was 21.0 (SD = 2.04) and all participants were right-handed. Their median familiarity with QWERTY layout was 5.5 (1: very unfamiliar; 7: very familiar). 4 users had prior experience with gesture typing. Their median familiarity with gesture typing was 6. We collected the participants' typing data, including the calibration parameters, the IMUs' attitude data, and the mapped coordinates series with the prototypical system described in section 4.3. The sample rate of the whole system is 200 FPS and the delay between the two IMUs is always within 25 ms (5 frames).

### 5.2 Design and Procedures

Similar to [51], DRG-Keyboard offers a typical indirect gesture typing solution where the input area (fingertip) and the display area (screen) are separated and the user inputs the trace with thumb-to-index fingertip gestures while looking at the real-time trace feedback on the screen to adjust their gesture dynamically so as to draw the desired trace. We researched two representative mental models - visual-dominated model and haptic-dominated model - for indirect typing in our study corresponding to different dominating factors and concentration levels. For the visual-dominated model, the visual clues served as the main guidance for gesture input, meaning the user paid attention to and relied more on the displayed trace to adjust their gesture. For the haptic-dominated model, the trace was drawn primarily based on the fingertip haptic feedback (e.g., the touching position and the moving direction) of the swiping gestures. The user less relied on the visual feedback, merely having a glance at the input traces to check the approximate correctness instead of staring at the trace, thus being able to input in a more relaxed manner. It is worth emphasizing that in both modes, users were well acknowledged of the principals of gesture typing that the gesture traces aim to form the desired shape instead of crossing every desired key precisely. To this point, the dominant factors merely affect how alike the trace looks as the user wanted.

After the participants signed the consent form, we first introduced the principles of gesture typing and the basic usage of DRG-Keyboard, including the mental model of the imaginary fingertip keyboard layout and the process to input a word, while helping the participants to put on the rings. Each ring is made of a flexible metal strip and the participants could adjust the tightness by bending the ends of the metal strip.

After putting on the rings, the participants were instructed to go through the calibration process. Then they were asked to examine whether the 2D coordinates were mapped correctly on a playground keyboard UI with visual feedback. After the calibration, we acknowledged the definitions of the two mental models to research, ensuring each participant understood the difference well. More concretely, we broke the definition of each model into multiple instructions as follows. For visual-dominated model, the user should 1) keep their gaze point on the cursor, 2) constantly control the movement of the cursor based on the mutual feedback of the fingertip haptics and the cursor trace, and 3) actively compensate the movement (e.g., by strengthening the effect of visual feedback in a behavior loop) once the cursor going in undesired direction or speed (based on the user's expectation, not necessarily identical with the target trace). For haptic-dominated model, they were instructed to 1) face to the screen with their eyes relaxed instead of fixing their gaze point to the cursor (or other widgets), 2) moving the cursor mainly based on the fingertip haptic feedback accompanied with occasional glance to ensure the approximate correctness, and 3) compensate the movement by casting more visual attention when they feel the drift between the current trace and their expected trace is unneglectable to cause difficulty or ambiguity to the algorithms.

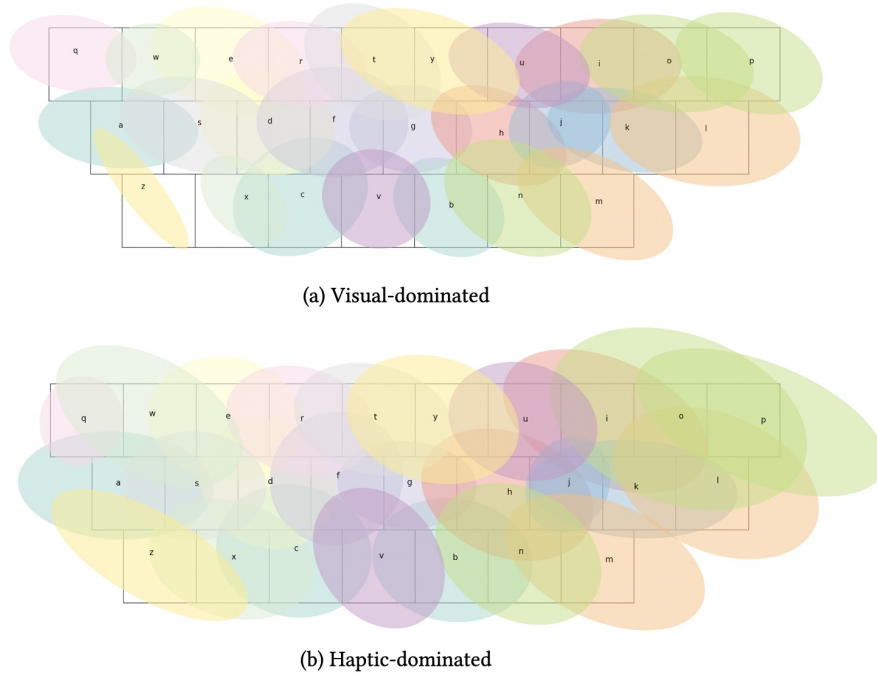


Fig. 10. Inferred touch point distribution on the screen keyboard (95% confidence ellipses) for the two mental models: a) visual-dominated and b) haptic-dominated.

Instructed by an interactive program, each participant was required to complete 2 sessions of input, corresponding to the two models - visual-dominated and haptic-dominated - as mentioned above. Each session consisted of 30 phrases randomly picked from the MacKenzie's phrase set [33]. To remove order effect, the order of the two sessions was shuffled. The participants input the words shown on the screen sequentially under the corresponding settings. After the experiment sessions, each participant was required to complete a questionnaire about their subjective perception, including mental load, physical load, effort, perceived accuracy, and perceived speed, regarding the two input settings. The whole procedure took about 60 minutes for each participant.

### 5.3 Results

Below we analyze the touch point distribution, the gesture speed, and the subjective perception to form a comprehensive understanding of thumb-to-index-fingertip gesture typing behavior in different mental models.

**5.3.1 Imaginary Key Distribution.** Unlike previous analysis of gesture typing [51], in which only the starting point and the ending point is assigned to the starting and ending letters in a word, we wanted to analyze the quality of the entire gesture trace instead of the endpoints. So we followed i'sFree [64] to locate the imaginary key position on a gesture trace using DTW algorithm [38]. We equidistantly sampled 100 points on the gesture trace and the template trace and applied DTW to obtain the optimal point-to-point assignment between the gesture points and the template points. For a vertex corresponding to a letter in a word, the centroid of the gesture points assigned to it is calculated as the imaginary key assigned to the letter.



Figure 10 shows the distributions of the imaginary key positions extracted from gestures for the two models. The scatter points of each imaginary key, approximately following a Gaussian distribution, are shown in different colors along with a 95% confidence ellipse and the corresponding letter at the centroid. The size of a key is 35 px × 35 px. The mean standard deviation (SD) of imaginary key positions was ( $\sigma_x = 16.48$  px or 0.47 KeyWidth,  $\sigma_y = 11.72$  px or 0.33 KeyWidth) for the visual-dominated model and ( $\sigma_x = 20.00$  px or 0.57 KeyWidth,  $\sigma_y = 15.03$  px or 0.43 KeyWidth) for the haptic-dominated model. It is within our expectation that the visual-dominated model yields a better point cloud with a lower SD both in the x-axis and the y-axis than the haptic-dominated model. For both models, the mean SD in the y-axis is smaller than that in the x-axis and the SDs in both axes are smaller than the keyboard size. The centroid of the point cloud of each letter for both models locates inside the corresponding key, showing that a user has a good perception of the imaginary key position and the thumb's relative movement on the index fingertip regardless of the dominant factor. We also found that the confidence ellipses in both settings have a systematic rotation offset (visual-dominated:  $24.31^\circ$  (SD =  $21.91^\circ$ ); haptic-dominated:  $19.37^\circ$  (SD =  $11.93^\circ$ )), meaning the errors in the x- and y- axes are potentially correlated.

**5.3.2 Gesture Speed.** The average elapsed time is 3.53 s for the visual-dominated model and 4.18 s for the haptic-dominated model. The average gesture speed (gesture length divided by elapsed time) is 6.09 *KeyWidth/s* (visual-dominated) and 4.99 *KeyWidth/s* (haptic-dominated) in the two settings respectively. It is not surprising the haptic-dominated model achieved higher gesture speed than the visual-dominated model because the latter requires additional effort to react to the visual feedback. We also observed that a user tends to draw shorter and smoother traces in the haptic-dominated model.

**5.3.3 Subjective Perception.** Figure 11 shows the subjective ratings in five aspects - mental load, physical load, perceived accuracy, perceived speed, and effort to achieve such performance, regarding the two mental models for indirect gesture input. Smaller values on mental load, physical load, and effort indicate users' better acceptance, while higher scores on speed and accuracy show higher perceived performance. A Wilcoxon Signed-Rank test shows that participants rated significantly lower mental load ( $Z = -2.11$ ,  $p < 0.05$ ), lower physical load ( $Z = -2.57$ ,  $p < 0.01$ ), lower effort ( $Z = -2.40$ ,  $p < 0.05$ ), and higher perceived speed ( $Z = -2.60$ ,  $p < 0.01$ ) for the haptic-dominated model compared to the visual-dominated model. The subjective ratings show that participants generally had better typing experience and faster input speed for the haptic-dominated model. For the perceived accuracy, despite the uncertainty of the input strokes caused by lower visual concentration for the haptic-dominated model, the difference is not significant ( $Z = -1.61$ ,  $p = 0.11$ ), meaning participants' confidence would not significantly decrease as they lowered their visual attention.

## 6 DRG-KEYBOARD: DECODING ALGORITHMS

In DRG-KeyBoard, the gesture decoding algorithm was designed based on the general elastic gesture decoder [26, 51, 58] and optimized based on the findings from Study 1. In this section, we explain 1) the principle of the general elastic gesture decoder and 2) the optimization strategy in our implementation to improve both the decoding speed and the decoding accuracy.

### 6.1 Elastic Matching for Gesture Recognition

To recognize a word gesture, we measure the similarity of the gesture with the templates of all the words in the dictionary. We first equidistantly sample  $N$  points  $\mathbf{g} = \{g_1, g_2, \dots, g_N\}$  on the gesture and  $\mathbf{t} = \{t_1, t_2, \dots, t_N\}$  on the target trace, then compute the similarity between the two point series  $\mathbf{g}$  and  $\mathbf{t}$ . Since direct point-wise mapping may lead global shift of the two point series, we use an elastic pattern matching algorithm [43], also known as dynamic time warping (DTW), to find the optimal point-to-point assignment. The distance between the gesture and the template can recursively calculated by a dynamic programming procedure

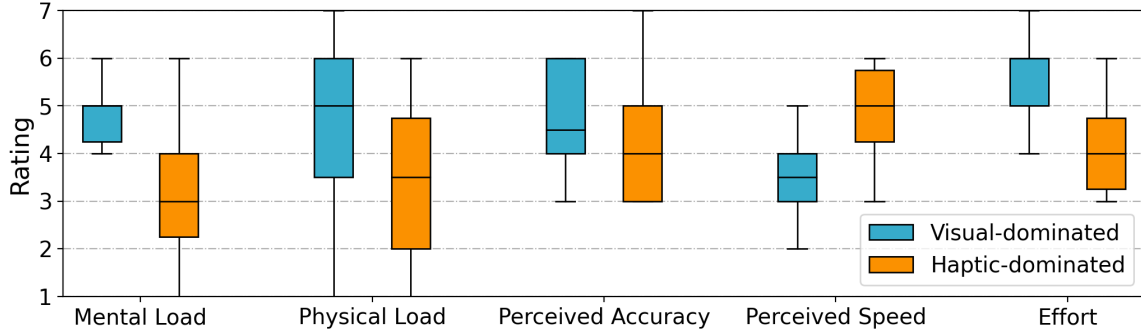


Fig. 11. Subjective ratings regarding the two mental models for indirect fingertip gesture typing in five aspects: mental load, physical load, perceived accuracy, perceived speed, and effort.

$$D(i, j) = d(g_i, t_j) + \min\{D(i, j - 1), D(i - 1, j), D(i - 1, j - 1)\} \quad (8)$$

where  $D(1, 1) = d(g_1, t_1)$  and  $d(\mathbf{g}, \mathbf{t}) = D(N, N)$  represents the distance between the two patterns.

## 6.2 Speeding up with Rigid Pruning

Since the computation complexity of DTW is  $O(N^2)$ , where  $N$  is the point number of a pattern, calculating the elastic distance between the gesture and each template in the distance is time-consuming. To speed up the calculation, we adopt a rigid matching algorithm pruning the templates with larger rigid distances. The rigid distance between the two patterns is the sum of the pair-wise distances of all the points

$$d(\mathbf{g}, \mathbf{t}) = \sum_{i=1}^N d(g_i, t_i) \quad (9)$$

with the computation complexity of  $O(N)$ . Therefore, the two-stage strategy - using rigid matching to fetch a subset of the original dictionary and applying elastic matching to the subset could speed up the calculation while taking advantage of the elasticity of DTW to get high-quality results.

## 6.3 Distance Measurement Optimization

Instead of directly using Euclidean distance for distance measurement, we expected to improve the decoding performance by applying certain transform over the original coordinates. The findings from Study 1 that users perceived the gesture more accurately in y-axis than in x-axis inspired us to adopt different weights on x-axis and y-axis. Thus we apply a scaling transform on the original coordinates based on the standard deviations in Study 1:

$$\Phi_{scaling}[(x, y)] = \left( \frac{x}{\sigma_x}, \frac{y}{\sigma_y} \right) \quad (10)$$

Further, as we observed in Study 1, the distribution of imaginary keys had a systematic rotation offset, a general linear transform may help to correct the systematic bias. Assuming the covariance matrix of the  $i^{th}$  key is  $cov_i$  (which is a symmetric matrix), we apply the singular value decomposition (SVD) on  $cov_i$  to acquire the transform matrix (or bases)  $M_i$  for the  $i^{th}$  key. We average all the matrices to acquire a global transform matrix  $M'$ , then

Table 3. The mean accuracies and standard deviations (%) regarding different distance transforms.

Transform	Top 1		Top 6		Top 20	
	Visual	Haptic	Visual	Haptic	Visual	Haptic
No Transform	46.11 (12.00)	37.58 (10.04)	73.12 (10.67)	66.55 (11.33)	83.48 (9.16)	79.83 (8.56)
Scaling	47.46 (12.12)	38.05 (11.03)	74.34 (12.08)	67.10 (11.76)	85.78 (9.38)	80.39 (9.57)
Linear	47.39 (12.06)	39.02 (11.46)	73.93 (12.53)	66.76 (11.80)	86.12 (9.74)	80.18 (10.28)
Scaling + Personalized	48.82 (12.76)	38.33 (10.33)	74.27 (11.78)	66.41 (11.51)	85.44 (9.67)	80.25 (9.02)

apply the inverse matrix to normalize the coordinates. The formula derivation is as below:

$$\begin{aligned}
 U_i^T, D_i, U_i &= SVD(cov_i) \\
 M_i &= U_i D_i^{\frac{1}{2}} \\
 M' &= \frac{1}{26} \sum_{i=1}^{26} M_i
 \end{aligned} \tag{11}$$

$$\Phi_{linear}[(x, y)] = M'^{-1}(x, y)^T$$

After applying a proper transform, either a scaling transform or a linear transform, we measure the Euclidean distance of two points in the post-transform coordinates:

$$d(g_i, t_i) = \|\Phi_*(g_i) - \Phi_*(t_i)\|_2 \tag{12}$$

Above we used the statistics from all users to build a global transform model. It's worth emphasizing that the statistics from an individual can be derived to build a personalized transform model that better fits the behavioral characteristics of the desired user.

#### 6.4 Simulation Results

We set the number of sample points  $N$  to 32 and ran the simulation of our 2-stage decoding algorithm on the data from Study 1. We incorporate the unigram and bigram language models derived from the Bayes' rule [16] in our simulation. The score of a word  $w$  given an input gesture  $g$  is computed as:

$$S(w|g; context) = S(w|g) \cdot P(w|context)^{-\alpha} = d(tr(w), tr(g)) \cdot P(w|context)^{-\alpha} \tag{13}$$

where  $P(w|context)$  is the probability of  $w$  from the desired language model (no LM, unigram, and bigram).  $tr(*)$  represents the sampled traces (points) of a gesture or a word template.  $S(w|g)$ , or  $d(tr(w), tr(g))$  is the spatial score between  $w$  and  $g$  given by our 2-stage decoding algorithm.  $\alpha$  is a parameter to control the strength of the language model.

In our simulation, we used a dictionary of 10,000 words, containing the most frequent words from the American National Corpus [1], which cover over 90% of written English words [35]. The top one million bigram frequency data from the Google Web 1T 5-gram database [7] smoothed by Katz's back-off model [22] were used in our bigram model.

Table 3 shows the simulation accuracy using different distance metrics - no transform, scaling transform, linear transform, and personalized scaling transform (corresponding to the transforms described in subsection 6.3) - for the two mental models. To evaluate the pure effect of distance transform, no language model is used in the simulation. The rigid pruning size is set to 100. Generally, applying scaling or linear transform leads to accuracy improvement in both settings and personalized transform further boosts the decoder's performance.

Table 4. The mean accuracies and standard deviations (%) regarding different sizes in rigid pruning.

Rigid Size	Top 1		Top 6		Top 20	
	Visual	Haptic	Visual	Haptic	Visual	Haptic
N = 50	44.28 (11.48)	37.37 (10.43)	69.06 (11.48)	65.00 (12.42)	78.27 (9.60)	76.81 (9.50)
N = 100	46.11 (12.00)	37.58 (10.04)	73.12 (10.67)	66.55 (11.33)	83.48 (9.16)	79.83 (8.56)
N = 200	47.05 (12.06)	38.05 (10.51)	74.88 (11.12)	67.52 (11.85)	86.80 (8.92)	81.76 (9.32)

Table 5. The mean accuracies and standard deviations (%) using different language models.

Language Model	Top 1		Top 6		Top 20	
	Visual	Haptic	Visual	Haptic	Visual	Haptic
No LM	47.39 (12.04)	38.47 (10.68)	75.90 (11.85)	67.31 (12.08)	87.54 (8.76)	81.42 (9.18)
Unigram	64.52 (10.53)	56.78 (11.35)	87.61 (8.03)	83.69 (9.22)	93.23 (5.46)	90.57 (7.13)
Bigram	69.67 (7.02)	65.24 (9.30)	87.34 (5.68)	84.58 (6.96)	92.28 (5.17)	90.02 (6.15)

An RM-ANOVA shows significant effect of *Transform* on the average accuracy ( $F_{3,57} = 3.53, p < 0.05$ ). However, the effect between *Scaling* and *Linear* is not significant ( $F_{1,19} = 0.09, p > 0.75$ ). This is probably due to the insignificance of the angular bias in the data.

Table 4 shows the simulation accuracy regarding different rigid pruning size. The transform is set to *Scaling* as described in subsection 6.3 and no language model is used in the simulation. Results show that increasing the rigid pruning size would constantly improve the decoding performance. The effect of rigid pruning size is significant on the average accuracy ( $F_{2,38} = 36.80, p < 0.001$ ). Note that the rigid pruning size directly indicates the complexity of the DTW, the average time consumption to decode a word is 0.16s, 0.29s, and 0.53s respectively, which is positively correlated with the rigid pruning size.

Table 5 shows the simulation accuracy using different language models. The transform is set to *Scaling* and the rigid pruning size is set to 200 in this simulation.  $\alpha$  is set to 0.1 based on a pilot study. We found that incorporating stronger language models brings significant improvement to the average accuracy ( $F_{2,38} = 153.36, p < 0.001$ ). A thrilling finding is that, the bigram language model tremendously improves the simulation accuracies of the haptic-dominated model, bringing the results competitive with the accuracies of the visual-dominated model (e.g., 65.24 v.s. 69.67; 84.58 v.s. 87.34; and 90.02 v.s. 92.28). This finding indicates that a user can perform gesture typing on the fingertip in a natural manner relying less on visual feedback to get a good result, demonstrating the superiority of DRG-Keyboard in usability.

## 7 STUDY 2: EVALUATING DRG-KEYBOARD'S PERFORMANCE

We conducted a user study to evaluate the performance of thumb-to-index-fingertip gesture typing with DRG-Keyboard. We were interested in understanding how well a user performs gesture typing with DRG-Keyboard.

### 7.1 Participants and Apparatus

We recruited 10 participants (2 females, all ESL) from the local campus. The average age was 20.4 (SD = 1.4) and all participants were right-handed. Their median familiarity with QWERTY layout was 6 (1: very unfamiliar; 7: very familiar). 5 users (P1, P2, P3, P4, P8) had prior experience with gesture typing and their median familiarity with gesture typing was 6. The apparatus was same as Study 1. In this study, we used the following setting for our decoder: rigid pruning (N=50) + scaling transform + bigram language model.

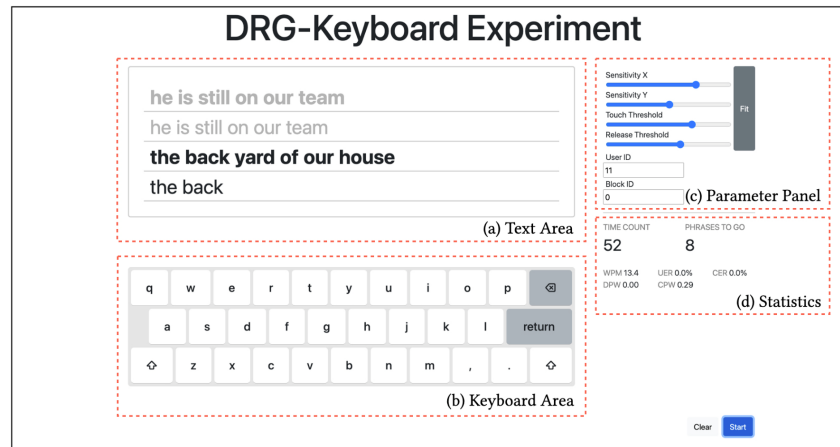


Fig. 12. The web UI used in Study 2, consisting of four main regions: text area, keyboard area, parameter panel, and statistics.

## 7.2 Design and Procedures

We devised an interactive web application to guide our experiment. As shown in Figure 12, the web UI consists of four main regions: text area, keyboard area, parameter panel, and statistics. During the experiment, the text area displays the words to be entered. The user can gesture-type in the keyboard area, adjust the keyboard parameters in the parameter panel, and see the real-time statistics in the statistics area.

Same as Study 1, the basic usage and mental model of DRG-Keyboard were first introduced to the participants. Then the participants were required to put on the rings as well as going through the calibration process (as described in Section 4.1 and 4.2). They were asked to examine whether the 2D coordinates were mapped correctly on the keyboard area with visual feedback. They were also required to adjust the parameters, including the sensitivities in two axes and the touch-down and touch-up thresholds, to fit their usage.

Prior to the formal experiment, participants were asked to practice for as long as they wanted, ensuring they were sufficiently familiar with the typing method and the input process. During the formal experiment, participants transcribed 4 blocks, each containing 10 distinct phrases randomly picked from the MacKenzie's phrase set [33]. Participants enter a word with the 2-stage method described in section 3.2 - performing a word gesture on the QWERTY keyboard followed by candidate selection on a pie navigation widget, as shown in Figure 3. Two phrases were always displayed on the screen, the bottom one is the one being transcribed. After a phrase having been transcribed, the next phrase is automatically displayed on the screen.

For comparison, we asked the participants to practice and take a 1-minute free-form gesture typing test on the smartphone before or after the formal experiment (the order was randomly picked to remove order effect) to provide a reference gesture typing speed. The participants were required to type as fast as possible to accomplish text transcription tasks on a commercial web APP (<https://www.livechat.com/typing-speed-test>) with the default iOS gesture keyboard.

Further, we noticed that 4 of the participants (P1, P2, P4, P8) had the experience of using the Swipe Keyboard on the latest Apple Watch Series 7, which was a competitive technique similar with DRG-Keyboard both in the form factor (e.g., gesture typing on the tiny interface) and the usage scenario (e.g., mobile texting). So we invited them to participate in an appending study to make a comprehensive comparison between DRG-Keyboard and Apple Watch Swipe Keyboard. The participants were first required to practice with the Apple Watch Swipe Keyboard and then complete a 10-phrase transcription task (the phrases were randomly picked from the MacKenzie's phrase set [33]) where their completion time and operation statistics were recorded. After the transcription

task, we conducted a semi-structured interview with the participant to compare the two techniques from the perspectives of experience, task load, usage scenarios, and form factor. Specifically, the questions we included in the interview were: 1) What are the advantages and disadvantages of DRG-Keyboard compared with Apple Watch Swipe Keyboard? 2) Which technique would you prefer more considering the form factor and why? 3) What kind of application scenarios do you think the two techniques fit for and what scenario would you prefer to use DRG-Keyboard?

The main study took around 60 minutes (15 minutes for practice, 40 minutes for DRG-Keyboard's formal experiment, and 5 minutes for smartphone gesture typing test) and the appending study took around 25 minutes (5 minutes for practice, 5 minutes for transcription, and 10 minutes for interview), depending on participants' input speed. We collected 400 phrases (10 participants  $\times$  4 blocks  $\times$  10 phrases) of DRG-Keyboard's typing records in the study.

### 7.3 Results

**7.3.1 Input Speed.** We measured the average input speed following MacKenzie's equation [32]

$$WPM = \frac{|S| - 1}{T} \times 60 \times \frac{1}{5} \quad (14)$$

where  $|S|$  is the length of the transcribed string, and  $T$  is the elapsed time in seconds.

Figure 13 shows the average gesture input speeds (error bars indicate 95% confidence interval) in the 4 blocks. Results from RM-ANOVAs showed that there was a significant effect of *Block* ( $F_{3,27} = 25.68$ ,  $p < 0.001$ ). The average speed in the first block was 9.9 WPM (SD = 3.4) and constantly increased to 12.9 WPM (SD = 3.7) in the last block, with an improvement of 30.3%. The learning curve did not converge in the last block. So the result does not reflect the upper bound.

Meanwhile, the average gesture typing speed of all participants on the smartphone was 18.9 WPM (SD = 3.6). The input speed in block 4 could achieve 68.3% of that on the smartphone. Moreover, the maximum input speed among all participants was 17.0 WPM (P4) in the first block and 20.1 WPM (P4) in the last block, which is competitive with the gesture typing speed on the smartphone. Although such a comparison was not able to be strictly controlled in the corpus, the decoder's capability, and the interaction modality (e.g., the selection mechanism, and the candidate number), we believed the result could demonstrate the efficiency and the potential of DRG-Keyboard because we were comparing our prototypical system with the default iOS keyboard which is a mature commercial system being well-optimized in the above factors.

Here we also established an empirical comparison between DRG-Keyboard and existing techniques with similar interaction modalities. As is most similar with DRG-Keyboard, TipText [50] implemented thumb-to-index-fingertip tap-based text entry with capacitive finger sleeves, reaching an average typing speed of 10.5(0.6) WPM in the first block and 13.3(0.5) WPM in the fourth block. Swipeboard [12] proposed a swipe-based typing method on an ultra-small interface (e.g., 12mm  $\times$  12mm), achieving an input speed of 9.1 WPM in the first block and 14.0 WPM in the fourth block (further increasing to 19.6 in the eighth block under 2 hours of dense practice). As a comparison, DRG-Keyboard achieved a competitive input speed (9.9(3.4) WPM to 12.9(3.7) WPM) and a similar learning curve with the two established techniques. We noticed that the average speed of DRG-Keyboard was a bit lower than that of TipText (11.5 WPM v.s. 11.9 WPM), probably due to the proficiency of the user group (e.g., all users from our study were ESL users and 5 of them had no experience on gesture typing, leading to a greater deviation in WPM) and optimization details (e.g., adapting a real-time decoding (e.g., stream DTW [39]) strategy and using an auto-complete algorithm [50]).

**7.3.2 Errors.** We reported the uncorrected error rate (UER), total error rate (TER), cancel-per-word (CPW), and delete-per-word (DPW) in this study. Uncorrected errors are the errors found in the final string, while total errors

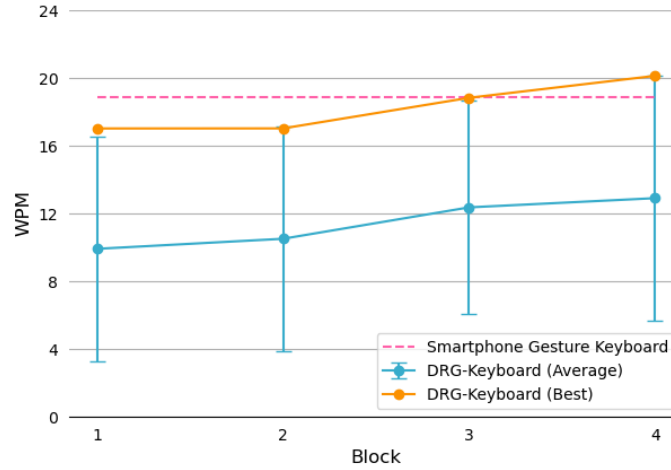


Fig. 13. Text input speed of DRG-Keyboard. The blue line refers to the average input speed. Error bars indicate 95% confidence intervals. The orange line refers to the maximum speed in each block and the pink dotted line represents the reference speed of smartphone gesture typing.

Table 6. Frequency of different word-affecting operations. (N = 2898)

Select						Cancel	Delete
1st	2nd	3rd	4th	5th	6th		
1817(62.7%)	219(7.6%)	91(3.1%)	55(1.9%)	19(0.7%)	23(0.8%)	613(21.2%)	61(2.1%)

include both corrected and uncorrected errors. Cancel-per-word and delete-per-word indicate how frequently a cancel operation or a delete operation happens.

Figure 14 (a) shows the UER and the TER in the 4 blocks and Figure 14 (b) shows the CPW and the DPW. Results from RM-ANOVAs show no significant effect of *Block* on *UER* ( $F_{3,27} = 1.942$ ,  $p = 0.147$ ), *TER* ( $F_{3,27} = 1.429$ ,  $p = 0.256$ ), *CPW* ( $F_{3,27} = 1.333$ ,  $p = 0.284$ ), and *DPW* ( $F_{3,27} = 1.897$ ,  $p = 0.154$ ). The average UERs of the 4 blocks are 3.0% (SD = 4.2%), 1.2% (SD = 1.6%), 2.0% (SD = 1.2%), and 3.8% (SD = 2.9%) respectively. Generally, participants had low UERs in the 4 blocks, indicating good performance in the text entry task. The mean TERs of the 4 blocks are 6.9% (SD = 6.0%), 4.2% (SD = 3.4%), 3.5% (SD = 2.6%) and 5.2% (SD = 2.3%) correspondingly. The average CPW of the whole study is 0.27 (SD = 0.16), which is consistent with the simulation results in Section 6.4. Users had a low DPW of 0.03 (SD = 0.04) in the whole study.

**7.3.3 Interaction Statistics.** Table 6 illustrated more details about how users performed word-affecting actions, more precisely, *Select*, *Cancel* and *Delete*. The most frequent operation was *Selecting* (76.7%), among which users mostly chose our Top-1 prediction (62.7%). This was consistent with the accuracy result in Section 6.4. For 21.2% of the operations, users *Cancelled* the words they had committed, probably because their desired word did not occur as a candidate. For the rest 2.1% of the operations, users *Deleted* the word for correction.

In most of the *Cancelling* cases (578 out of 613, 94.3%), our decoder did not report the desired words as Top-20 candidates. It might be attributed to ill-shaped traces caused by touch detection errors or IMU drifting. In the meanwhile, a relatively low hit rate for Top7-20 (1.2%) compared with that for Top6 (76.7%) validated our 6-word-presented confirmation design.

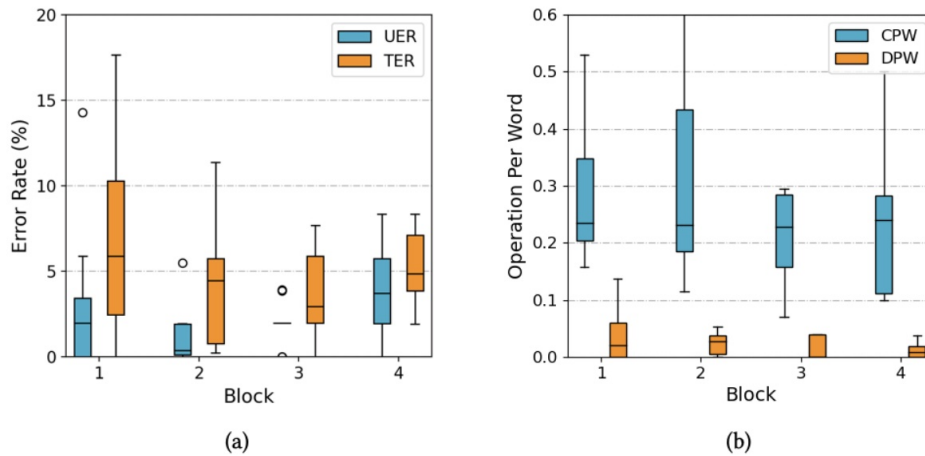


Fig. 14. Statistics of errors: (a) uncorrected error rates (UERs) and total error rates (TERs) over blocks; (b) cancel-per-words (CPWs) and delete-per-words (DPWs) over blocks.

**7.3.4 Subjective Feedback.** Below we summarized participants' subjective comments. Some participants (P1, P3, P5, P7) showed a positive attitude towards the decoder and got overwhelmed during the experiment when our decoder precisely predicted the desire word with a track which in their perspective was out of shape. Comments like *"It's unbelievable that I can perform text typing just on the fingertip, and I am in good control of the cursor's movement"*(P4) also proved the good haptic feedback of DRG-Keyboard. There were also a few users getting frustrated at the beginning of the experiment (P8, P9), feeling hard to map their finger movement to 2-D coordinates and find a proper threshold to detect touch and release. These negative feelings were reduced with the proceeding of the experiment as they got more familiar with our device.

**7.3.5 Comparison with Apple Watch Swipe Keyboard.** Participants who took part in the appending study had an average input speed of 16.2 (SD=3.9) WPM with 5.5% (SD=2.0%) TER and 5.1% (2.2%) UER on the last block in the main study. Their average typing speed with Apple Watch Swipe Keyboard was 20.0 (SD=3.2) WPM with 13.5% (5.1%) TER and 1.8% (1.9%) UER. The input speed of DRG-Keyboard was approximately 80% of that on Apple Watch Swipe Keyboard. Regarding the error, we found the users tended to input more errors (e.g., TER 13.5% v.s. 5.5%) and corrected most of them (e.g., CER 11.7% v.s. 0.4%) with Apple Watch Swipe Keyboard rather than with DRG-Keyboard, indicating smaller correction cost for Apple Watch Swipe Keyboard according to the aversion modeling of typing errors [4, 5]. Such statistics were understandable because the user could simply click a button, attributed to the touchscreen interface, and retype the word for the Apple Watch Swipe Keyboard while they should perform a clicking gesture to delete the word and experienced a two-stage input process with fair delay (also, most errors were captured at the "cancel" phase for efficiency) with DRG-Keyboard, leading the user to type in a faster and free manner using the Apple Watch Swipe Keyboard.

We summarized interview results and user comments as follows. *Advantages and drawbacks comparing DRG-Keyboard and smart watch (Q1).* Advantages for DRG-Keyboard included typing in natural manners (P2, P4), privacy guaranteed by subtle behaviours (P2, P4), and single-hand interaction (P1, P2, P4, P8). P4 added that *"self-haptic feedback provided by thumb movement on index fingertip brings the awareness of absolute position, while I would gradually lose control of position if using a smart watch without visual attention."* On the other hand, participants reported that limited screen size for smart watch increased false touches and thus required more attention when selecting or deleting words (P1 and P2). As for the drawbacks against smartwatches, P1 and



P8 showed their concern about the current typing speed. Physical load caused by long time finger usage (P2) and mental load (P8) were also mentioned.

*Form factors (Q2).* Regarding the form factor, all the participants leaned on DRG-Keyboard for the reasons of one-handedness (P1, P2, P4, P8), pervasiveness (e.g., always-available and seamless switching across devices, P2), less visual attention and hand occupation (P2), comfort (P1), privacy-preserving (P4), and more suitable for HMDs and IoT screens (P8). As for the current form of mobile texting where the interface was displayed on the watch, P8 thought typing directly on the watch (e.g. replying to a quick message) was an intuitive and convenient choice. However, all participants agreed the tiring two-handed typing form (raising the arm and touching the watch screen with the other hand) impeded them from heavy texting on the watch, while DRG-Keyboard had a more friendly form.

*Different suitable usage scenarios for these two techniques (Q3).* P1 and P8 favored smartwatch technique in typing tasks requiring higher efficiency and accuracy considering current typing performance. P1 noted that it would be more comfortable to have arms supported (by desktop) when typing on smartwatch. P2 thought typing on the watch was kind of intermediate form between smartphone and ring, which had narrowed application scenarios - "If my hands were fully available, I would prefer my smartphone for heavy typing tasks. If not, neither can I use the watch". On the other hand, DRG-Keyboard is more welcome for light and quick typing tasks (P8), or other scenarios where users prefer their motions not being limited (e.g. during walking) (P1, P2, P4, P8). P2 also pointed out given the visual resource was distributed in the environment and always-available (e.g., for AR, VR, IoT, and large screen scenarios), DRG-Keyboard would be a better choice.

## 8 LIMITATIONS AND DISCUSSION

In this section, we discuss the limitations as well as the potential considerations related to the practical adoption and deployment of DRG-Keyboard.

### 8.1 Form Factor and Keyboard Design

Currently, the form factor (e.g., where the rings are worn on) and the keyboard design (e.g., the typing area and the keyboard layout) are determined based on a pilot brain-storming interview. Although the current design of DRG-Keyboard was well accepted by the users, it is also worthwhile to investigate more on the possibilities of form, input modality, and design. For example, what happens if we only use one ring, or attach the rings to different segments of the fingers? Is it possible to adapt the keyboard to the tip of the middle finger? Are a touch-down and thumb-to-finger friction essential for a fingertip gesture keyboard? Can we use other keyboard layouts instead of QWERTY for gesture input? All these questions are valuable and worth in-depth research in the future.

### 8.2 Optimization of Touch Detection

In our work, touch-up and touch-down events are recognized based on a simple combination of the relative attitude data and the accelerometer's frequency-domain data, allowing the user to adjust the thresholds during usage for self-adaption. The evaluation of this algorithm shows that our touch detection algorithms work pretty well in our typing scenario. We envision the sensing can be further improved for in-the-wild touch detection. To this point, the current version may not work well in some extreme cases, e.g., repeatedly touching in high frequency. There are two potential directions to optimize the touch detection algorithm. The first is to leverage machine learning models, such as SVM, Random Forest, and CNN, to recognize free-form touching events robustly based on vast training data. The second is to introduce additional hardware, such as an RF circuit, to detect subtle on-body contact. By such optimizations, a user would have a better experience and stronger capability to enable touch-based interactions in the wild.

### 8.3 Eyes-Free Gesture Typing for Small Vocabulary

The current version of DRG-Keyboard is designed for gesture typing on a large vocabulary with visual feedback. We envision DRG-Keyboard has the potential to support eyes-free gesture typing at a fast speed. Our informal study shows eyes-free fingertip gesture input can reach approximately half of eyes-on's decoding accuracy. This suggests that gesture typing on the fingertip eyes-freely is likely to be achieved by reducing the dictionary size and optimizing the decoding algorithms. Such a modality is useful in command input and fast navigation in various scenarios.

### 8.4 Out-of-Vocabulary Input and Alternative Keyboard Layouts

Like many text entry techniques with inaccurate input [19, 50, 51], the original version of DRG-Keyboard does not support OOV (Out of Vocabulary) words. However, an inherited advantage of DRG-Keyboard is that, with a simple modification, DRG-Keyboard can support OOV input with almost no additional adapting cost. Results from Study 1 show that users generally have a good perception of relative finger movement, being able to reach a precise key on the keyboard. To this point, a simple but effective method to support OOV input is that a user types one character at each stroke. In the OOV mode, the dictionary of the decoder is changed to the desired charset. With such a modification, a user can easily enter punctuations, names, and numbers without changing the input behavior.

Similarly, due to users' good perception and control capability of the trace, switching to another keyboard layout (e.g., the T9 layout) almost requires no additional effort (but just changing the UI and the corresponding dictionary). Although DRG-Keyboard showed the effectiveness of gesture typing on a fingertip QWERTY keyboard, it remained an open question what is the optimal layout for fingertip gesture typing in naturalness and efficiency, which is worthy of further research. From the other perspective, such a layout-adapting feature would benefit broad application scenarios like numerical keyboard, wheel menu, and other forms of customized menus.

### 8.5 Application Scenarios

As validated in the paper, DRG-Keyboard offers the unique capability of subtle touch and 2D relative cursor input on the fingertip, whose superiority for real-world applications is three-fold:

**8.5.1 Text entry for restricted hand movement.** The most direct advantage of DRG-Keyboard is that the user could type in a subtle manner under scenarios where the hand movement is restricted (e.g., with one hand occupied), such as commuting with one hand holding the handbag, photographing with one hand holding the camera, and attending a meeting. Text entry in these scenarios emphasizes more on subtleness and privacy rather than typing speed, thus is most suitable for DRG-Keyboard. Moreover, since DRG-Keyboard requires less visual attention than direct input techniques (e.g., Apple Watch Swipe Keyboard), it is possible for the user to text as a parallel task while maintaining their major attention on the main task (e.g., sending a short message in the lecture without omitting the content).

**8.5.2 Temporal combination with ring-based interaction.** Since DRG-Keyboard shares similar sensing schemes with DualRing [29], an inherited advantage was to combine DRG-Keyboard's text entry capability with DualRing's broad interaction space in the temporal domain, bringing strong input capability. For example, the user could use a pinching gesture (on different fingers) to switch the target input device or UI widget seamlessly. Also, combined with the UI control capability of DualRing (e.g., simulating cursor and discrete buttons), the user could freely operate 2D and 3D interfaces with text input on the fingertip (which is a thrilling capability).

**8.5.3 Reconfigurability.** As proposed by Sections 8.3 and 8.4, DRG-Keyboard has the potential to support different input modalities and different keyboard layouts due to its high reconfigurability, based on which various applications can be designed. For example, a reduced dictionary could probably support blind typing for command

and navigation while the modification of the dictionary and keyboard layout could support special functionalities such as OOV input and numeric input.

## 9 CONCLUSION

We present DRG-Keyboard, a miniature QWERTY gesture keyboard on the fingertip supported by dual IMU rings. Benefit from the strong sensing capability to detect subtle relative hand movement and the touch-up and touch-down events, DRG-Keyboard supports subtle gesture typing on the fingertip with perfect haptic feedback. Our user study analyzed users' fingertip gesture typing behavior with DRG-Keyboard, based on which we optimized the general elastic gesture decoder while demonstrating the efficiency of DRG-Keyboard through a performance evaluation study. Our research serves an important role in the following two aspects: 1) we leverage an indirect sensing technique to sense micro fingertip gestures without impeding self haptic feedback; and 2) we enable gesture keyboard on an extremely subtle fingertip gesture space. We envision our work would play an important role in future input techniques for mobile, wearable, and AR devices and open new doors for the design of micro fingertip gesture interaction.

## ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No.62132010, Beijing Academy of Artificial Intelligence (BAAI), 2025 Key Technological Innovation Program of Ningbo City under Grant No.2022Z080 and National Key R&D Program of China under Grant No.2020AAA0105200. Our work is also supported by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, and Institute for Artificial Intelligence, Tsinghua University (THUIAI).

## REFERENCES

- [1] 2021. American National Corpus. <http://www.anc.org>
- [2] Sunggeun Ahn, Seongkook Heo, and Geehyuk Lee. 2017. Typing on a Smartwatch for Smart Glasses. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) (ISS '17). Association for Computing Machinery, New York, NY, USA, 201–209. <https://doi.org/10.1145/3132272.3134136>
- [3] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. NENYA: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 2043–2046. <https://doi.org/10.1145/1978942.1979238>
- [4] Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K. Dey, and Jennifer Mankoff. 2017. Quantifying Aversion to Costly Typing Errors in Expert Mobile Text Entry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 4229–4241. <https://doi.org/10.1145/3025453.3025695>
- [5] Nikola Banovic, Ticha Sethapakdi, Yasasvi Hari, Anind K. Dey, and Jennifer Mankoff. 2019. The Limits of Expert Text Entry Speed on Mobile Keyboards with Autocorrect. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) (MobileHCI '19). Association for Computing Machinery, New York, NY, USA, Article 15, 12 pages. <https://doi.org/10.1145/3338286.3340126>
- [6] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual Gesture Keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (UIST '12). Association for Computing Machinery, New York, NY, USA, 137–146. <https://doi.org/10.1145/2380116.2380136>
- [7] Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Ver. 1.
- [8] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. <https://doi.org/10.1145/2807442.2807450>
- [9] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) (UIST '13). Association for Computing Machinery, New York, NY, USA, 255–260. <https://doi.org/10.1145/2501988.2502016>
- [10] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. UTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) (UIST '13). Association for Computing Machinery, New York, NY, USA, 237–244. <https://doi.org/10.1145/2501988.2502035>

- [11] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 1504–1514. <https://doi.org/10.1145/2858036.2858125>
- [12] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-Small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 615–620. <https://doi.org/10.1145/2642918.2647354>
- [13] Kiran Dandekar, Balasundar I. Raju, and Mandayam A. Srinivasan. 2003. 3-D Finite-Element Models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense. *Journal of Biomechanical Engineering* 125, 5 (10 2003), 682–691. <https://doi.org/10.1115/1.1613673> arXiv:[https://doi.org/10.1145/2642918.2647354](https://arxiv.org/abs/https://doi.org/10.1145/2642918.2647354)
- [14] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 161–166. <https://doi.org/10.1145/2642918.2647396>
- [15] Barrett Ens, Ahmad Byagowi, Teng Han, Juan David Hincapié-Ramos, and Pourang Irani. 2016. Combining Ring Input with Hand Tracking for Precise, Natural Interaction with Spatial Analytic Interfaces. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (Tokyo, Japan) (SUI '16). Association for Computing Machinery, New York, NY, USA, 99–102. <https://doi.org/10.1145/2983310.2985757>
- [16] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (CHI '15). Association for Computing Machinery, New York, NY, USA, 649–658. <https://doi.org/10.1145/2702123.2702503>
- [17] Masaaki Fukumoto and Yasuhito Suenaga. 1994. "FingeRing": A Full-Time Wearable Interface. In *Conference Companion on Human Factors in Computing Systems* (Boston, Massachusetts, USA) (CHI '94). Association for Computing Machinery, New York, NY, USA, 81–82. <https://doi.org/10.1145/259963.260056>
- [18] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 1059–1070. <https://doi.org/10.1145/3332165.3347947>
- [19] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 128 (Dec. 2020), 29 pages. <https://doi.org/10.1145/3432204>
- [20] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwipe: Word-Gesture Typing Using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300244>
- [21] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 1526–1537. <https://doi.org/10.1145/2858036.2858483>
- [22] S. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35, 3 (1987), 400–401. <https://doi.org/10.1109/TASSP.1987.1165125>
- [23] Wolf Kienzle and Ken Hinckley. 2014. LightRing: Always-Available 2D Input on Any Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 157–160. <https://doi.org/10.1145/2642918.2647376>
- [24] Junhyeok Kim, William Delamare, and Pourang Irani. 2018. ThumbText: Text Entry for Wearable Devices Using a Miniature Ring. In *Graphics Interface*. Toronto, Canada. <https://hal.archives-ouvertes.fr/hal-03030524>
- [25] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-Based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) (UIST '04). Association for Computing Machinery, New York, NY, USA, 43–52. <https://doi.org/10.1145/1029632.1029640>
- [26] Per-Ola Kristensson and Shumin Zhai. 2005. Relaxing Stylus Typing Precision by Geometric Pattern Matching. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (San Diego, California, USA) (IUI '05). Association for Computing Machinery, New York, NY, USA, 151–158. <https://doi.org/10.1145/1040830.1040867>
- [27] Chandan Kumar, Ramin Hedeshy, I. Scott MacKenzie, and Steffen Staab. 2020. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376317>
- [28] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (UIST '16). Association for Computing Machinery, New York, NY, USA, 321–333. <https://doi.org/10.1145/2984511.2984582>

- [29] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 115 (sep 2021), 27 pages. <https://doi.org/10.1145/3478114>
- [30] Guan hong Liu, Yizheng Gu, Yiwen Yin, Chun Yu, Yuntao Wang, Haipeng Mi, and Yuanchun Shi. 2020. Keep the Phone in Your Pocket: Enabling Smartphone Operation with an IMU Ring for Visually Impaired People. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 58 (June 2020), 23 pages. <https://doi.org/10.1145/3397308>
- [31] Christian Lochair and Sean Gustafson. 2010. PinchWatch: A Wearable Device for One-Handed Microinteractions.
- [32] I. Scott MacKenzie. 2015. A Note on Calculating Text Entry Speed. <http://www.yorku.ca/mack/RN-TextEntrySpeed.html>
- [33] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (*CHI EA '03*). Association for Computing Machinery, New York, NY, USA, 754–755. <https://doi.org/10.1145/765891.765971>
- [34] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-Air Word-Gesture Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (*CHI '14*). Association for Computing Machinery, New York, NY, USA, 1073–1082. <https://doi.org/10.1145/2556288.2556964>
- [35] Paul Nation and Robert Waring. 1997. Vocabulary size, text coverage and word lists. *Vocabulary: Description, acquisition and pedagogy* 14 (1997), 6–19.
- [36] Yue Qin, Chun Yu, Zhaoheng Li, Mingyuan Zhong, Yukang Yan, and Yuanchun Shi. 2021. ProxiMic: Convenient Voice Activation via Close-to-Mic Speech Detected by a Single Microphone. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 8, 12 pages. <https://doi.org/10.1145/3411764.3445687>
- [37] Gulnar Rakhmetulla and Ahmed Sabbir Arif. 2021. SwipeRing: Gesture typing on smartwatches using a segmented QWERTY around the bezel. In *Proceedings of the 2021 Graphics Interface Conference, GI*, Vol. 21.
- [38] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.
- [39] Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. 2007. Stream Monitoring under the Time Warping Distance. In *2007 IEEE 23rd International Conference on Data Engineering*. 1046–1055. <https://doi.org/10.1109/ICDE.2007.368963>
- [40] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-Available Input with Muscle-Computer Interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) (*UIST '09*). Association for Computing Machinery, New York, NY, USA, 167–176. <https://doi.org/10.1145/1622176.1622208>
- [41] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 59 (June 2020), 25 pages. <https://doi.org/10.1145/3397309>
- [42] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (Tokyo, Japan) (*ISS '18*). Association for Computing Machinery, New York, NY, USA, 177–187. <https://doi.org/10.1145/3279778.3279799>
- [43] C. C. Tappert. 1982. Cursive Script Recognition by Elastic Matching. *IBM Journal of Research and Development* 26, 6 (1982), 765–771. <https://doi.org/10.1147/rd.266.0765>
- [44] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016. TouchRing: Subtle and Always-Available Input Using a Multi-Touch Ring. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) (*MobileHCI '16*). Association for Computing Machinery, New York, NY, USA, 891–898. <https://doi.org/10.1145/2957265.2961860>
- [45] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. 2016. ThumbRing: Private Interactions Using One-Handed Thumb Motion Input on Finger Segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) (*MobileHCI '16*). Association for Computing Machinery, New York, NY, USA, 791–798. <https://doi.org/10.1145/2957265.2961859>
- [46] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. 2017. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). Association for Computing Machinery, New York, NY, USA, 3095–3105. <https://doi.org/10.1145/3025453.3025704>
- [47] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (Sept. 2017), 21 pages. <https://doi.org/10.1145/3130978>
- [48] Mathias Wilhelm, Daniel Krakowczyk, Frank Trollmann, and Sahin Albayrak. 2015. ERing: Multiple Finger Gesture Recognition with One Ring Using an Electric Field. In *Proceedings of the 2nd International Workshop on Sensor-Based Activity Recognition and Interaction* (Rostock, Germany) (*iWOAR '15*). Association for Computing Machinery, New York, NY, USA, Article 7, 6 pages. <https://doi.org/10.1145/2761111.2761117>

- [//doi.org/10.1145/2790044.2790047](https://doi.org/10.1145/2790044.2790047)
- [49] Zheer Xu, Weihao Chen, Dongyang Zhao, Jiehui Luo, Te-Yen Wu, Jun Gong, Sicheng Yin, Jialun Zhai, and Xing-Dong Yang. 2020. BiTipText: Bimanual Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376306>
- [50] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 883–899. <https://doi.org/10.1145/3332165.3347865>
- [51] Zhican Yang, Chun Yu, Xin Yi, and Yuanchun Shi. 2019. Investigating Gesture Typing for Indirect Touch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 117 (Sept. 2019), 22 pages. <https://doi.org/10.1145/3351275>
- [52] Yui-Pan Yau, Lik Hang Lee, Zheng Li, Tristan Braud, Yi-Hsuan Ho, and Pan Hui. 2020. How Subtle Can It Get? A Trimodal Study of Ring-Sized Interfaces for One-Handed Drone Control. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 63 (June 2020), 29 pages. <https://doi.org/10.1145/3397319>
- [53] Hui-Shyong Yeo, Wenxin Feng, and Michael Xuelin Huang. 2020. WATouCH: Enabling Direct Input on Non-Touchscreen Using Smartwatch's Photoplethysmogram and IMU Sensor Fusion. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3313831.3376198>
- [54] Hui-Shyong Yeo, Xiao-Shen Phang, Steven J. Castellucci, Per Ola Kristensson, and Aaron Quigley. 2017. Investigating Tilt-Based Gesture Keyboard Entry for Single-Handed Text Entry on Large Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 4194–4202. <https://doi.org/10.1145/3025453.3025520>
- [55] Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi, and Yuanchun Shi. 2017. Word Clarity as a Metric in Sampling Keyboard Test Sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 4216–4228. <https://doi.org/10.1145/3025453.3025701>
- [56] Xin Yi, Chun Yu, Weinan Shi, and Yuanchun Shi. 2017. Is it too small?: Investigating the performances and preferences of users when typing on tiny QWERTY keyboards. *International Journal of Human-Computer Studies* 106 (2017), 44–62.
- [57] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture? Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 4479–4488. <https://doi.org/10.1145/3025453.3025964>
- [58] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (CHI '03). Association for Computing Machinery, New York, NY, USA, 97–104. <https://doi.org/10.1145/642611.642630>
- [59] Shumin Zhai and Per Ola Kristensson. 2012. The Word-Gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101. <https://doi.org/10.1145/2330667.2330689>
- [60] Shumin Zhai, Per Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the Iphone: From the Laboratory to the Real World. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA) (CHI EA '09). Association for Computing Machinery, New York, NY, USA, 2667–2670. <https://doi.org/10.1145/1520340.1520380>
- [61] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E. Starner, Omer T. Inan, and Gregory D. Abowd. 2017. FingerSound: Recognizing Unistroke Thumb Gestures Using a Ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 120 (Sept. 2017), 19 pages. <https://doi.org/10.1145/3130985>
- [62] Tengxiang Zhang, Xin Zeng, Yinshuai Zhang, Ke Sun, Yuntao Wang, and Yiqiang Chen. 2020. *ThermalRing: Gesture and Tag Inputs Enabled by a Thermal Imaging Smart Ring*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376323>
- [63] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu, and Yuanchun Shi. 2018. Forceboard: Subtle text entry leveraging pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–10.
- [64] Suwen Zhu, Jingjie Zheng, Shumin Zhai, and Xiaojun Bi. 2019. I'sFree: Eyes-Free Gesture Typing via a Touch-Enabled Remote Control. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300678>