# PrinType: Text Entry via Fingerprint Recognition

ZONGJIAN LIU, Department of Automation, BNRist, Tsinghua University, China
JIELING HE, Department of Automation, BNRist, Tsinghua University, China
JIANJIANG FENG*, Department of Automation, BNRist, Tsinghua University, China
JIE ZHOU, Department of Automation, BNRist, Tsinghua University, China

174

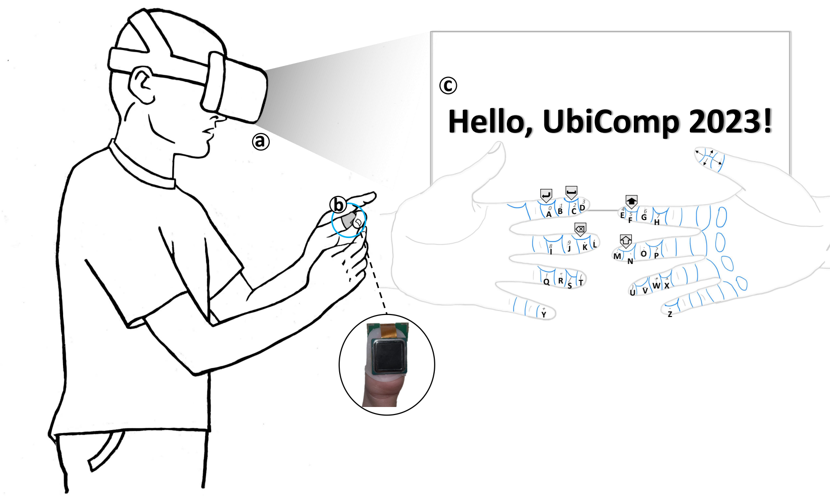



Fig. 1. PrinType enables rich text input for (a) VR HMDs by using (b) a small thumb-worn sensor to recognize different finger regions assigned to different keys, and (c) a virtual keyboard with highlighted keys via the display of HMDs.

We present PrinType, a fingerprint recognition based typing technique for virtual reality. Different regions of fingers covered by friction ridges are assigned to different keys (i.e. letters, numbers, punctuation, or functions). Once the thumb-worn fingerprint sensor touches a finger, the contact region (and its key) is recognized by matching the current image with registered templates. Using only a small sensor, PrinType turns each segment of all fingers into a touchable key. We designed keyboard layouts corresponding to three interaction sub-spaces: whole finger keyboard, fingertip keyboard, and single-handed keyboard.

*Corresponding author

Authors' addresses: Zongjian Liu, liu-zj20@mails.tsinghua.edu.cn, Department of Automation, BNRist, Tsinghua University, Beijing, China; Jieling He, hejl19@mails.tsinghua.edu.cn, Department of Automation, BNRist, Tsinghua University, Beijing, China; Jianjiang Feng, jfeng@tsinghua.edu.cn, Department of Automation, BNRist, Tsinghua University, Beijing, China; Jie Zhou, jzhou@tsinghua.edu.cn, Department of Automation, BNRist, Tsinghua University, Beijing, China.


2474-9567/2022/12-ART174
https://doi.org/10.1145/3569491

A 12-person user study was conducted to evaluate the performance of different strategies. Our user evaluation showed that participants achieved an average of 29.56, 32.38, and 34.22 WPM with 0.79%, 0.20%, and 0.21% not corrected error rate in the three strategies. In addition, we provided a detailed analysis of various micro metrics to further understand user performance and technical characteristics. Overall, PrinType is favored by users for its usability, efficiency, and novelty.

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

Additional Key Words and Phrases: finger touch input, ridge-based interaction, text entry, wearable devices

**ACM Reference Format:**
Zongjian Liu, Jieling He, Jianjiang Feng, and Jie Zhou. 2022. PrinType: Text Entry via Fingerprint Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 174 (December 2022), 31 pages. https://doi.org/10.1145/3569491

## 1 INTRODUCTION

With the increasing popularity of Virtual Reality (VR) Head-Mounted Displays (HMDs), interest in using VR for remote work and office work is growing [54]. Compared with traditional 2D monitors, VR HMDs are a big leap with respect to display and are considered to be the ultimate display devices. However, input devices for VR still lag far behind in terms of efficiency and flexibility [12]. Existing consumer-grade immersive HMD VR systems often rely on indirect control of a virtual pointer using handheld controllers [51, 66, 80]. While it is sufficient for occasional input (such as user names, passwords), it is far from convenient for heavy text entry required in office and communication applications [20].

Physical keyboards are natural solutions for heavy text input in VR, which have been widely studied in VR to bank on their familiarity [6, 21, 22, 32, 38, 49]. Unable to see the keyboard and hands, some researchers [49] proposed blending video of the keyboard and hands into VR, while others chose to track the keyboard and/or hands with various techniques to display their virtual representations in VR. These methods generally support fast and rich text input, however, physical keyboards require a flat surface as support and are too bulky to carry around, limiting the scenarios of VR text input. Imagine a possible scenario where a VR user lying on the couch or sitting on the bus has to reply to a message. Some researchers explored mid-air typing on a virtual QWERTY keyboard enabled by hand tracking techniques [13, 22, 78]. However mid-air typing may cause fatigue [31] and seem obtrusive in the public [33]. In addition, precise marker-based hand tracking techniques are expensive and not mobile [13, 22], while bare hand tracking methods are not sufficiently robust or accurate due to occlusion and unideal illumination [78]. Therefore, physical keyboards or virtual keyboards enabled by external tracking devices may not be the optimal choice for text entry in various VR scenarios.

In contrast, hand-worn devices implemented by various sensors do not need supporting surface or bulky external devices, and have received much attention of text entry researchers. A popular interaction method is thumb-to-finger tapping at different locations of the fingers that serve as virtual keyboards [9, 35, 36, 40, 59, 75, 77]. This interaction method does not require visual attention thanks to human natural perception of the relative position between the thumb and other fingers. Based on the positions where the sensors are placed, hand-worn typing methods can be divided into finger-worn devices or wrist-worn devices. Finger-worn devices are mostly capacitive sensors, covering one finger [36, 77] or more [35, 40, 76]. Tiny sensors with small area are preferred due to good portability, but it is easy to mistype due to high-density keys. Hence, statistical language models are usually needed to reduce word ambiguity in these methods. Wrist-worn sensors mostly detect the positions where the thumb taps, which have larger input area and do not affect the freedom of other fingers. But it is still challenging to ensure accurate finger tracking in fast finger typing [58, 73].

In this paper, we present PrinType, a fingerprint recognition based typing technique for virtual reality. With a miniature fingerprint sensor attached to the user's thumb finger, as shown in Figure 1, PrinType can identify the position where the sensor touches by matching current image with registered templates. Different regions of fingers are assigned to different keys, constituting a fully-functional keyboard in virtual reality. Thanks to the flexibility of the fingers and the robustness of fingerprint recognition technology, almost all the palmar side of fingers can be touched and recognized, which is important for accommodating the large number of keys contained in a typical physical keyboard. Because the small sensor is mounted only on one thumb, the contradiction between performance and convenience faced by capacitive touch typing method is avoided [40, 77]. With the sensitive perception of fingertips and real-time feedback in VR, the problem of keyboard invisibility is alleviated.

Based on the recognition results in a pilot study, we validate that the recognizable finger regions of different participants are similar, which constitute the available keys for keyboard layout design. In terms of these regions, we designed a fully-functional virtual keyboard for interaction along with three alternative text entry strategies. The strategies are divided into three interaction sub-spaces [41]: hand-to-hand interaction, fingertip interaction of two hands, and finger interaction on the same-sided hand (SSH) [75]. All strategies employed layouts based on alphabetical order for the main keyboard, supplemented by two switching keyboards, flexibly adjusted by two shift keys. In addition to uppercase and lowercase letters, our keyboards also support numbers, punctuation, symbols, and common function keys, e.g. space, backspace, enter, and four arrow keys. The first strategy is a full keyboard, while the latter two strategies reduce the number of keys sharply. A statistical decoder is used to predict the target word efficiently in all strategies.

We conducted a 12-person user study to evaluate the performance of PrinType and compared it with a baseline method, the physical keyboard. Simple sentences and complex sentences (not just lowercase letters) were used to test the overall performance of PrinType across the three strategies and the baseline method. The learnability and robustness of PrinType were also tested in the user study. Study results showed that participants achieved 29.56, 32.38, and 34.22 words per minute using PrinType with different strategies for the simple sentence test. The typing performance in different humidity conditions demonstrated the robustness of the recognition algorithm. In terms of user preference, most of the participants preferred the third strategy. In addition to investigating entry speed and error rates in different strategies, the detailed recording data allows us to explore micro metrics of performance in virtual reality.

The main contributions of our work are three-fold:

(1) We propose and implement PrinType, which uses fingers as the interactive surface and fingerprint recognition as sensing technique, to enable efficient typing in virtual reality.

(2) We design three alternative typing strategies and realize rich text entry functionality with switching keyboards.

(3) We evaluate the performance of PrinType through user study with both simple and complex sentences to simulate typing in VR office. We also analyze underlying micro metrics of performance among different users and typing strategies to further explore the performance potential.

## 2 RELATED WORK

Most typing methods in VR are hand-based and the rest are based on head pointing, eye gazing, voice transcription, etc. Hand-based methods are divided into typing with hand-worn devices and external devices. Since PrinType enables typing with a hand-worn device, hand-based typing methods are mainly reviewed in this section. Table 1 summarizes hand-based typing methods in VR with their entry performance. The table also includes information about the decoder, the supported character set, and visual feedback. Note that only methods with quantitative experiments are summarized in the table.

### 2.1 Hand-based Typing with Hand-worn Devices

*2.1.1 Glove-based Devices.* Using gloves as interactive sensing techniques is a feasible way to realize text input. Rosenberg and Slater [62] proposed a chording-based glove with finger sensors and shift buttons, designing it as

Table 1. Hand-based text entry methods in VR with their sensing techniques, entry speed in words per minute (WPM), and error rate (%). ('$\sim$' in Performance represents the approximate value, due to different measurement indicators in the paper.) Information about the decoder, supported character set and visual feedback is also included.

| **Sensing Technique** | **Method** | | | **Employ Decoder** | **Support Full Character Set** | **Visual Feedback** | | **Entry Performance** | |
|---|---|---|---|---|---|---|---|---|---|
| | Authors | Year | Abbreviation | | | Display Layout | Real-time Keystroke | Entry Speed (WPM) | Error Rate (%) |
| Glove | Kuester et al. | 2005 | KITTY [39] | ✗ | ✓ | ✗ | ✓ | 1.8 - 5 | 57 |
| | Whitmire et al. | 2017 | DigiTouch [74] | ✗ | ✗ | ✓ | ✓ | 16 | 0.85 |
| Wrist-worn Device | Pratorius et al. | 2014 | DigiTap [58] | ✗ | ✗ | ✗ | ✗ | $\sim$10 | 6.3 |
| | Wang et al. | 2015 | PalmType [73] | ✗ | ✗ | ✓ | ✓ | 10.01 | 1.58 |
| | Streli et al. | 2022 | TapType [68] | ✓ | ✗ | ✗ | ✗ | 17.1/18.0/19.2 | 0.5/0.3/0.6 |
| Finger-worn Device | Xu et al. | 2019 | TipText [77] | ✓ | ✗ | ✗ | ✗ | 13.3 | 0.30 |
| | Gupta et al. | 2019 | RotoSwype [27] | ✗ | ✗ | ✓ | ✓ | 14 | $\sim$1 |
| | Jiang et al. | 2019 | HiFinger [35] | ✗ | ✓ | ✓ | ✓ | 9.82 | 4.14 - 7.48 |
| | Xu et al. | 2020 | BiTipText [76] | ✓ | ✗ | ✗ | ✗ | 23.4 | 0.03 |
| | Lee et al. | 2021 | FingerText [40] | ✗ | ✗ | ✓ | ✓ | 22.38 - 31.3 | 3 - 12 |
| | Zhang et al. | 2022 | TypeAnywhere [85] | ✓ | ✗ | ✓ | ✗ | 41.6 - 70.6 | 0.95 - 1.71 |
| Physical Keyboard | Gruber et al. | 2018 | - [21] | ✗ | ✓ | ✓ | ✓ | $\sim$36 | 5 - 15 |
| | Otte et al. | 2019 | - [55] | ✗ | ✓ | ✓ | ✓ | $\sim$32 | $\sim$1.5 |
| Touchscreen or Touchpad | Gugenheimer et al. | 2016 | FaceTouch [25] | ✗ | ✓ | ✓ | ✓ | $\sim$10 | 0.5 |
| | Lu et al. | 2017 | BlindType [44] | ✓ | ✗ | ✓ | ✗ | 17 - 23 | $\sim$1.7 |
| | Boustila et al. | 2019 | - [5] | ✗ | ✗ | ✓ | ✓ | 8.98/12.10/21.42 | 1.90/1.12/2.30 |
| | Bai et al. | 2021 | - [2] | ✗ | ✓ | ✓ | ✓ | $\sim$10 | $\sim$20 |
| VR Controller | Yu et al. | 2018 | Circular PizzaText [80] | ✗ | ✗ | ✓ | ✓ | 8.59 - 15.85 | 1.59 |
| External Camera | Hoppe et al. | 2018 | qVRty [32] | ✗ | ✓ | ✓ | ✓ | $\sim$31 | 12.03 |
| | Fashimpaur et al. | 2020 | PinchType [15] | ✗ | ✗ | ✓ | ✓ | 12.54 | $\sim$1.5 |

a text entry device with full ASCII character set input. Bowman et al. [7] detected finger pinch on Pinch Gloves with conductive cloth sewn into the tips of each of the fingers to achieve typing. KITTY [39] and DigiTouch [74] utilize electronic touch points or partially conductive fabric strips on the gloves to detect finger touch events, enabling complex finger typing interactions. Mobile Lorm Glove [18] translates the hand-touch alphabet Lorm into text for the deafblind. Argot [56] is a one-handed wearable glove with 15 buttons, enabling text-entry with weak magnetic interaction as haptic feedback. However, these typing modalities require users to wear gloves, affecting how they interact with everyday objects, which is not ideal in terms of flexibility and comfortableness.

*2.1.2 Wrist-worn Devices.* Another wearable text entry methods are wrist-worn devices equipped with various sensors. PalmType [73] uses a wrist-worn IR sensor to detect users' finger position and enables typing by tapping on the palm. BlueTap [11] and DigiTap [58] map the letters onto the fingers in an alphabetic order, and use a wrist-worn camera to detect the taps. Accurate detection of tapping position using wrist-worn cameras is difficult due to occlusion and fast movement of fingers.

ViFin [10] is a mid-air continuous micro finger writing method using a wrist-worn smartwatch with the Inertial measurement unit (IMU) to detect vibration from the movement of users' index fingers. Deep network and spell check are exploited to decode continuous characters of index finger writing, which is computationally complex. Besides, the recognition accuracy of ViFin [10] is relatively low. TapType [68] uses two wireless wristbands with accelerometers, first introduced in TapID [50], to sense mechanical vibrations arising from finger taps on flat surfaces. Bayesian tap classification and probabilistic text entry decoder are employed to estimate the character sequences of ten-finger typing. As a learning-based tapping finger identification approach, TapType [68] needs to train the classifier in a supervised manner, which may have problem with generalization on different users and surface materials. In addition, the input process is slow for out-of-vocabulary words.

*2.1.3 Finger-worn Devices.* Finger-worn devices are flexible and lightweight, enabling expressive text input interaction by sensing rotation/touching events [17, 24, 27, 30, 35, 36, 40, 52, 75–77] and recognizing hand gestures

[26, 82, 83]. For example, FingeRing [17] generates key code based on the combination of tapping events of fingers with accelerometers and enables typing on any surface such as the waist or thigh. Similarly, TypingRing [52] and QwertyRing [24] measure the movement of fingers by different sensors and identify the selected keys of an invisible keyboard on any desk-like surface. TypeAnywhere [85] uses two Tap Straps and an accelerometer-based finger-worn device for tap detection on any surface and it exploits deep network to decode typing sequences based on finger-tap sequences outputted by Tap Straps. TapGazer [30] detects tapping fingers in ten-finger typing on any surface using various sensors and then disambiguates by letting users select target words with gaze. RotoSwype [27] uses a 6-axis motion-tracking sensor and a velcro ring with a physical button for word-gesture typing. Users rotate the hand so that the pointer traces a path over letters and click the ring's button to end input, and then the top predicted word will be outputted. It is also possible to type simple words by gesture recognition with finger-worn sensors like accelerometer, gyroscope, and microphone [26, 82, 83].

Methods based on hand gestures [26, 82, 83] do not have sufficient keys for fast typing, so these methods are not suitable for complex text input. Ring-form devices that sense key events based on acceleration motion information [17, 24, 30, 52] or hand rotation ranges [27] typically require auxiliary sensing information to ensure accurate typing. The operation of text input in these methods needs to be very precise, which is not intuitive and hard to learn for beginners in a VR setting. TapGazer [30] combines finger taps and gaze to type in VR, which requires extra attention and may lead to fatigue after extended use. In addition, most of the above techniques employ statistical decoder for improving typing speed and lack efficient input method for large symbol set and out-of-vocabulary words.

Besides typing by detecting motion information of fingers, text input methods by sensing on-body touch events are widely studied, especially in VR. TipText [77], BiTipText [76] and ThumbText [36] feature a miniature touch-sensitive keyboard on fingers and input a character after a two-step touch event. FingerText [40] and FingerT9 [75] explore one-handed text entry by intra-hand touches between the thumb and fingers with capacitive sensing technique.

In the above touch sensor based techniques, keys are mapped to different regions on fingers, palms, or fingernails. These methods take full advantage of the flexibility of fingers [34]. However, the input area of finger typing is usually significantly restricted (otherwise it will have the drawback of gloves), thus limiting the number of keys [36, 40, 77], which will result in inefficienct typing. In fact, few of them support full character set input. To this end, we utilize fingerprint sensors amounted to a thumb to identify different regions of fingers, significantly expanding the input space and meanwhile not affecting fingers to perform other tasks.

## 2.2 Hand-based Typing with External Devices

*2.2.1 Physical Keyboards.* Physical QWERTY keyboards or its variants were adapted to VR to take advantage of their familiarity and suitability for heavy text input tasks [6, 21, 22, 32, 38, 49, 57, 71]. Various forms of feedback have been studied to lower error rates and to improve entry speed. Optical tracking techniques are used to track physical keyboards and hands for providing informative feedback in [6, 13, 22, 32, 38, 47, 49, 67]. Less informative feedback, such as highlighting pressed keys, was also found useful for improving typing performance [21, 71]. Keypads and chording keyboards with fewer keys were tested in VR [19], achieving relatively low typing speeds.

Physical keyboards are too bulky to carry around and typically require a flat surface, which is not always available or convenient in VR scenarios. In addition, many of these studies require keyboard and hand tracking techniques for better typing performance, which can be expensive, hard to install, or not reliable.

*2.2.2 Touchscreen or Touchpad.* Text entry on touchscreen or touchpad brings familiar mobile phone interaction into virtual reality [2]. BlindType [44] explores eyes-free typing on a touchpad using one thumb, wherein a user taps on an imaginary QWERTY keyboard while receiving text feedback on a separate screen. Bai et al. [2], Boustila et al. [5], and Kim et al. [37] track the hands on smartphones and enable the users to type in VR by aligning the virtual interface with their own smartphones. Gugenheimer et al. [25] leverage the backside of the HMD as a touch-sensitive surface and users can type on a split QWERTY virtual keyboard by selecting the target that is touched when lifting the finger from the surface.

However, these methods often require additional large size touchscreen or touchpad as input devices, which will lead to inconvenience in VR applications. PrinType uses fingers as a natural virtual keyboard, detaching the layout from the screens. Besides, typing methods on touchscreen or touchpad are based on visual feedback, so they require extra attention to keep track of the current state, which are not intuitive and effective compared to haptic feedback between fingers in PrinType.

*2.2.3 Handheld Controller.* Hand-held controllers are used for text entry in many consumer-grade virtual reality systems. Speicher et al. [66] compared typing with handheld controller pointing, controller tapping, freehand and head pointing. Some methods divide the virtual keyboard into multiple regions in order to facilitate multi-key selection. For example, PizzaText [80] used a circular layout for VR using the two thumbsticks of a game controller, while Min [51] presented a text input tool for immersive VR by exploiting 3×3 screen cells and users can point the key indirectly by pointing its corresponding cell. Typing with handheld controllers is popular since the controller is always available and it is easy to learn. However, handheld controller is not suitable for long-term text input in office applications.

*2.2.4 External Camera.* Some methods use body-worn or external cameras to recognize the movements or gestures of the hands, so as to realize typing in VR. TypeNet [46] realized fast typing on a flat surface in front of a camera. Yi et al. [78] and Dudley et al. [13] evaluated ten-finger freehand mid-air typing using leap motion and optical tracking, respectively, and empirically investigated users' mid-air typing behavior. These methods [13, 46, 78] require two or ten fingers to type on a virtual QWERTY keyboard, continuing the users' typing habits. OmniTouch [29] is a shoulder-worn depth-sensing and projection system that enables interactive text entry on everyday surfaces like palms.

Sridhar et al. [67] and Fashimpaur et al. [15] explored text entry in mid-air by recognizing different gestures of the hands, which rely on leap motion and Optitrack motion capture system to track the fingers. In these methods, each hand gesture will be recognized as one or multiple letters. Inspired by word-gesture keyboard in smartphones [81], Markussen et al. [47] proposed Vulture, a mid-air word gesture typing technique, where a marker-based optical tracking system is used to track hands and fingers. In Vulture, the best matching word is outputted based on the trace of the pinch in the air.

The aforementioned typing methods make text entry more flexible by removing physical keyboards. Nevertheless, it is demonstrated that (1) users are possibly limited in their speed by errors in tracking under fast motion [67], and (2) mid-air typing usually results in fatigue after extended use [31, 74]. Since camera based finger tracking is still a challenging problem, most of these studies used expensive optical tracking techniques, making them impractical. In contrast, our typing technique recognizes the character based on analyzing the high resolution contact image captured by a cheap fingerprint sensor, which is more robust and efficient.

## 2.3 Others

Various sensors embedded in typical HMDs can measure the movement of head and eyes and record voice. Recent advances in speech recognition have increased the popularity of voice transcription as a text entry method in different computing platforms, which however is inconvenient or socially inappropriate on certain occasions [63]. Yu et al. [79] and Speicher et al. [66] compared head-pointing based text input technique with other selection-based techniques, and Ma et al. [45] and Rajanna et al. [61] studied gaze typing in VR. In these methods [45, 61, 66, 79], however, the user needs to move the cursor to the target position, and then select the character by dwelling on it for a predetermined amount of time, resulting in fatigue when typing.

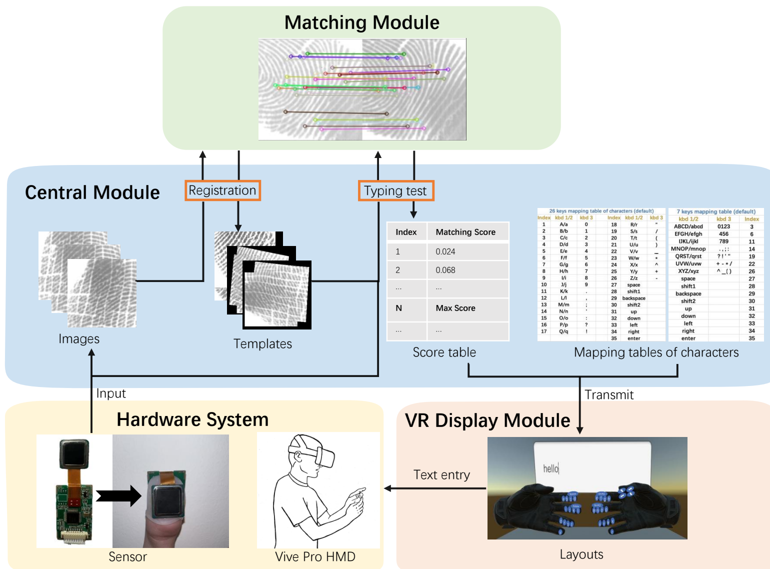



Fig. 2. The overall workflow of PrinType. The control end of PrinType includes the central module, the matching module, and the VR display module. Users can type in the selected layout of PrinType after registration, during which the central module will receive the input fingerprint image from the sensor in real time. Matching commands are then sent to the matching module to complete 1:N searching, which returns with the best matching index. The index, the keyboard layout, along with mapping table of characters is transmitted to the VR display module to update the display of characters, and provide visual feedback through HMD simultaneously.

## 3 PRINTYPE: HARDWARE AND ALGORITHM

In this section, we describe the technical details of PrinType, including the hardware and software implementation of the system.

### 3.1 Technical Overview

The overall workflow is shown in Figure 2. PrinType consists of a fingerprint sensor worn on the thumb and a head-mounted display (HMD) for typing task in the VR system. The control end of PrinType includes three modules: the central module, the matching module, and the VR display module.

In order to type in PrinType, a user needs to first select a keyboard layout and then register one's specified fingerprints according to the keyboard with the sensor in the central module. Fingerprint templates registered in sequence are stored in the local space exclusive to the user for matching with the input fingerprints during typing. The selected keyboard layout, along with information of each successful match will be transmitted to the Unity script in the VR display module. The total data transmission time is within 10 ms to ensure fast typing. Finally, the mapped characters from the fingerprint matching module will be displayed through HMD, and the corresponding keys of the specific layout in the scene will also be highlighted as visual feedback. The following technical details are divided into four dimensions: hardware system, fingerprint recognition algorithm, statistical language decoder and VR interface display.

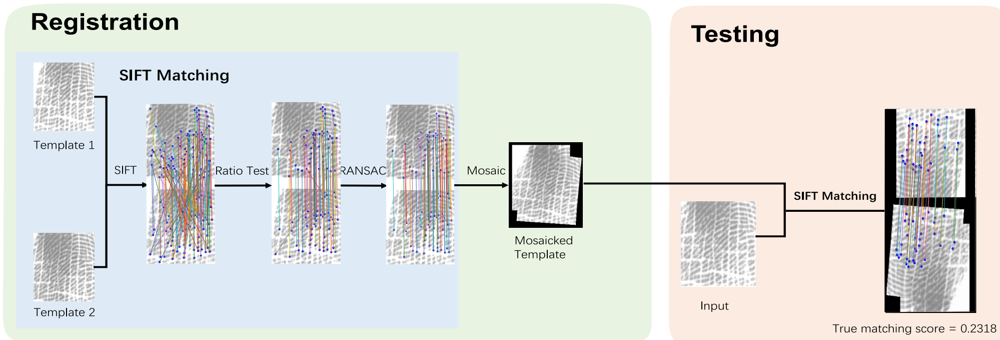



Fig. 3. Recognition process of PrinType. The process consists of two steps: registration and testing. The core recognition module, SIFT matching, is used to mosaic templates during registration, and calculate matching scores during testing.

## 3.2 Hardware System

The hardware system of PrinType mainly consists of VR HMD and capacitive fingerprint sensor. In addition, recognizable patterns distributed in different regions of user's fingers and palms will naturally form the virtual keyboard of PrinType.

The prototype of PrinType uses HTC Vive Pro [60] as VR device for Unity development. We embed 3D hand models and different keyboard layouts into the view of HMD to provide a complete typing system without affecting users' VR interaction.

The sensor is developed based on FPC1020AM, which can provide 500 ppi fingerprint images in real time. The active sensing size of the FPC1020AM is less than 1 $cm^2$ and weighs less than 1 gram, making it a lightweight sensor. Its module package mainly includes a transient voltage suppressor (TVS) for electrostatic discharge (ESD) protection purposes, and RC filters. In addition, the sensor is easily integrated using a few simple software commands sent through the high-speed Serial Peripheral Interface (SPI). The fingerprint image is delivered from the sensor in a raw unprocessed format, giving the user the opportunity to further process the image. Only a few additional low-cost, passive components are necessary to integrate the FPC1020AM sensor into a system, so it is ideal for mobile device development and optimization. The sensor can work properly in a wide range of temperature and humidity, and stably deliver high quality fingerprint images, which provides a basis for the robustness research of our typing system. This particular issue is investigated in more detail later in section 5.2.

## 3.3 Fingerprint Recognition Algorithm

Due to the requirement of typing characters accurately in real time, the fingerprint recognition algorithm should be able to recognize small fingerprint images as quickly and accurately as possible. We compared several keypoint extraction and matching methods using a small dataset of fingerprint images collected by PrinType [4, 8, 42, 43], and finally adopted the SIFT matching algorithm, which can meet the basic requirements of accurate real-time output. The SIFT algorithm detects blob-like keypoints instead of fingerprint minutiae and extracts gradient information around keypoints as descriptors. For this reason, almost every region of the hands, including even the joints, can be efficiently distinguished and recognized, providing more options for layout design. We designed three default layouts, as discussed in section 4, from which users need to select one before the registration.

The recognition process is shown in Figure 3. During the registration phase, the users are required to register each region three times. The preprocessing of enrolled templates consists of four consecutive steps: (1) Original

SIFT is used to detect feature points, and then match them in pairs into $m$ pairs. (2) Ratio test [43] is first applied to filter the matching points. It will reject the matches in which the distance ratio of closest to second-closest neighbors is greater than 0.8. (3) Then random sample consensus (RANSAC) [16] is adopted to further remove the false matching points, which is an iterative method to estimate parameters of a spatial transformation model. (4) Suppose there are $n$ matching pairs in the end, the matching score is computed as $\frac{n}{m}$. If the matching score exceeds the threshold, a simple fingerprint image mosaic is performed based on the transformation matrix calculated by the matching of SIFT feature points. Eventually, each finger region generates a mosaicked registration template for subsequent fingerprint matching. The template, which has been mosaicked twice, has larger area and more features. Therefore, fingerprints can be accurately matched when typing even if the pressed regions only partially overlap with the registration regions.

Once the templates are registered under the selected layout, users can flexibly modify the default mapping output for each template, as shown in Figure 2. For example, users can adjust their frequently used punctuation to more comfortable and recognizable finger regions, forming a personalized keyboard.

When typing in a specific layout, the input will be matched with every registered template in the gallery. A simple algorithm for searching an ideal matching index is to sort all the matching scores, and compare the top two most similar scores. If (1) the difference between the top two scores is large enough, and (2) the absolute value of the top score is above the threshold $\mu$, the template index corresponding to the top score will be output; otherwise, there will be no output. More formally, the $k$-th highest matching score $s_k$ in the registered template gallery $G$ for an input fingerprint $I$ is formulated as:

$$s_k(I,G) = Rank_k\left(\left\{s\left(I,C_g\right) \mid C_g \in G\right\}\right), \tag{1}$$

where $C_g$ is the candidate template index in the gallery $G$, $s\left(I,C_g\right)$ returns the matching score with template $C_g$, and $Rank_k\left(\cdot\right)$ returns the $k$-th highest score in the set. The output template index $O(I)$, if any, can be simplified as:

$$O(I) = \begin{cases} C\left(s_1\left(I,G\right)\right), & \text{if } \frac{s_1(I,G)}{s_2(I,G)} > \lambda, \text{and } s_1\left(I,G\right) > \mu \\ -1, & \text{otherwise} \end{cases}, \tag{2}$$

where $C\left(s\right)$ is the candidate template index corresponding to the score $s$.

The selection of parameters $\lambda\ (\lambda \geq 1)$ and $\mu\,(\mu > 0)$ is equivalent to a problem of expected risk minimization. As displayed in Equation (3), we measured the time cost of different choices with the possibility of $P\left(\cdot\right)$, assuming that the corrected error rate is zero: (1) If the output conditions are satisfied and the result is correct, the time cost is the minimum, which is the unit time of one pressing. (2) If the output conditions are satisfied but the result is wrong, the user needs to press backspace and type again with time costing to the maximum of 3 units. (3) If the system rejects to output, the user needs to retype regardless of true negative or false negative, so it costs 2 units of time. To sum up, the expected risk of typing time $R$ is calculated as:

$$\begin{aligned} R\left(\lambda,\mu\right) = {} & P(correct, \frac{s_1\left(I,G\right)}{s_2\left(I,G\right)} > \lambda, s_1\left(I,G\right) > \mu) \\ & + 2 * P(\left(\frac{s_1\left(I,G\right)}{s_2\left(I,G\right)} \leq \lambda\right) \cup \left(s_1\left(I,G\right) \leq \mu\right) \\ & + 3 * P(incorrect, \frac{s_1\left(I,G\right)}{s_2\left(I,G\right)} > \lambda, s_1\left(I,G\right) > \mu), \end{aligned} \tag{3}$$

and the ideal parameters are selected as:

$$\lambda^*,\ \mu^* = \underset{\lambda,\mu}{\arg\min}\ R\left(\lambda,\mu\right). \tag{4}$$

Table 2. Mean recognition Precision, Recall and F1-score of all participants in different finger regions.

| | Recall (%) | Precision (%) | F1-score (%) |
|---|---|---|---|
| **Fingertip** | 96.35 | **92.34** | **94.30** |
| **Belly** | **97.46** | 90.61 | 93.91 |
| **Sides of fingers** | 91.27 | 88.81 | 90.02 |
| **DIP** | 92.22 | 91.65 | 91.93 |
| **Second phalange** | 93.45 | 91.49 | 92.46 |
| **PIP** | 83.93 | 88.97 | 86.38 |
| **Others** | 83.44 | 91.45 | 87.26 |

Based on the analysis of prior studies, we selected $\lambda$ as 1.152, $\mu$ as 0.0348. We expect that through the reasonable selection of threshold values, the False Acceptance Rates (FAR) and the False Rejection Rates (FRR) of the fingerprint recognition can be best balanced.

### 3.4 Statistical Language Decoder

In order to reduce the error rate and improve the typing speed, a statistical language decoder is used to auto-correct the input sequence at the character level and provide a list of words for users to select. The statistical decoder in PrinType is modified according to the decoder in TapType [68]. The simplified decoder in PrinType does not use an additional word-level n-gram model because the raw recognition rate using fingerprints alone is already high enough (around 95% Rank 1 recognition rate in Figure 11). See Appendix A for a detailed description of the decoder.

### 3.5 VR Interface Display

On the user end, HTC Vive Pro hosts a complete VR scene of an interactive typing system. Specifically, we developed scenes based on Unity3D on SteamVR platform to achieve interactive virtual reality experience. In the 3D scene with HMD perspective as Main Camera, as shown in Figure 2, we embedded two 3D hand models as 3D Sprites to provide visual feedback of PrinType. The InputField on the Canvas behind the hand models is used for character input and display. Sprite elements labeled with characters are pre-created on all available positions of the 3D hands. Once the VR Display Module receives the keyboard layout information and mapping table of characters, the valid positions in the layout and the Sprite elements of the corresponding characters will be initialized to display. During the testing phase, words, phrases, or sentences to be typed are displayed in sequence on the Canvas. When the user presses a region on one’s hand, the Sprite element of the corresponding position will change its color to provide real-time feedback. When typing words, candidate words are displayed on the Canvas in real time and users select the target word by pressing the space key continuously.

## 4 KEYBOARD DESIGN

In this section, we discuss the layout design and rich typing function design of the virtual keyboard. We expect the keyboard design to be convenient, flexible, and suitable for people with different typing habits. Considering the differences in recognition, we designed a pilot study to better understand the design of the keyboard layouts. Based on the results of the pilot study and user survey, typing strategies are divided into three interaction sub-spaces, and all strategies employ layouts based on alphabetical order. Besides, we introduce rich typing function design in detail, such as continuous pressing function and switching keyboards function.

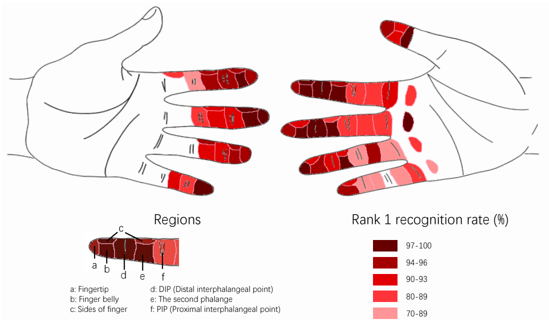



Fig. 4. Visualization of mean Rank 1 recognition rate in different regions of hands. The fingerprint sensor is attached to the left thumb. Fingers are divided into 65 fine-grained regions and the index of the top score is considered as the test output. Regions with higher recognition rates are rendered in darker colors. Rank 1 recognition rates of fingertips and finger bellies are generally over 95%, which will be used as an important reference for layout design.

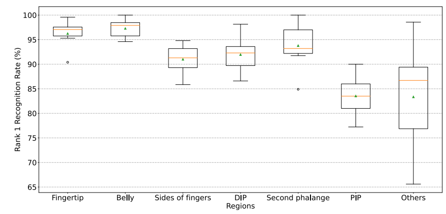



Fig. 5. Boxplot of mean Rank 1 recognition rate in different regions. In this boxplot, the dots indicate outliers based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$. Compared with fingertips and bellies, PIP and other regions have lower recognition rates, and the performance in these finger regions varies greatly across different users.

## 4.1 Pilot Study

On the premise that the sensor is attached to the left thumb of the participant, we selected 65 fine-grained divided candidate regions of the hands by taking into account the convenience of pressing and quality of ridge pattern, as displayed in Figure 4. We conducted a pilot study with 8 participants (4 females) to understand the effect of recognition performance on the layout design. Firstly, we asked each participant to register three times with the sensor in all 65 regions. Secondly, participants were instructed to press on each region in succession for five times. Finally, we surveyed participants about their preferences for typing layouts. The entire pilot study lasted less than 15 minutes for each participant.

During the test, an input fingerprint of a participant will only be compared to one's own fingerprint templates, and 65 matching scores are outputted. Assuming that the index of the top score is taken as the predicted result of each recognition, and the recognition Recall, Precision and F1-score of all finger regions of each participant can be obtained. Suppose region $a$ is pressed $n$ times, then the times of correct Rank 1 recognition is $TP$, and $FN = n - TP$. The times of other regions incorrectly recognized as region $a$ is FP. Thus the evaluation metrics of region $a$ are calculated as:

$$Recall = Rank\ 1\ Recognition\ Rate = \frac{TP}{TP + FN}, \tag{5}$$

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{7}$$

among which Rank 1 recognition rate directly reflects the overall recognition performance of one region. By visualizing the average Rank 1 recognition rate of all participants in each region, as shown in Figure 4, it can be seen that regions with high recognition rates usually include the fingertip, the finger belly, DIP (distal interphalangeal point), and the second phalange, possibly due to more feature information in these finger regions. In addition, regions where users cannot press fully will generate templates with large blank area, leading to poor recognition performance. Mean recognition results of different evaluation metrics in different finger regions are

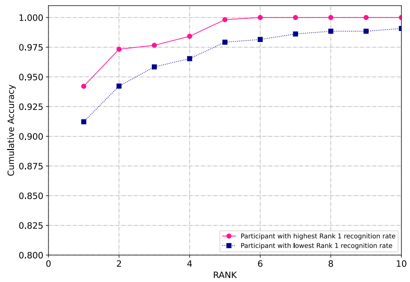



Fig. 6. Cumulative Match Characteristic (CMC) curves of the participant with highest Rank 1 recognition rate, and the participant with lowest Rank 1 recognition rate. The results show that all participants have good recognition performance and there is no significant difference among all participants. The mean Rank 1 recognition rate for all regions of all users is more than 0.88.

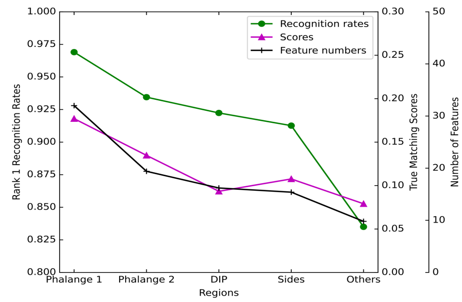



Fig. 7. Line charts of feature points, true matching scores and Rank 1 recognition rates varying with regions. Phalange 1 includes the tips and bellies of the finger (region a and b in Figure 4), and Phalange 2 refers to region e. Regions with more matching features during registration usually have better recognition rates and higher true matching scores during test.

summarized in Table 2, from which we noticed that the fingertip and the belly have good recognition performance in all evaluation metrics. As demonstrated in Figure 5, it can be concluded that finger regions with high Rank 1 recognition rates are similar across participants.

To compare the recognition performance across participants, we analyzed the recognition results of the participant with highest Rank 1 recognition rate, and the participant with lowest rate. Figure 6 compares the total CMC curves of the two participants. Cumulative Match Characteristic (CMC) curve is an indicator to evaluate pattern recognition system. Rank $N$ recognition accuracy refers to the proportion of query fingerprints that have true matching in the $N$ highest matching scores of all registration templates in the gallery. Slight differences across participants, not significantly, can be noticed in overall finger recognition from these curves.

During registration, if the matching scores of the fingerprints on the same position are lower than the threshold, the system will reject to generate templates and prompt the user to register again. One of the purposes of repeated registration is to reject fingerprints of poor quality and register fingerprints of good quality. Since the number of matching features between registered fingerprints can reflect the quality of fingerprints to a certain extent, it can also predict the recognition performance in typing test. We compared the number of matching features during registration, the average true matching scores and the Rank 1 recognition rates in the pilot study in Figure 7. The curve further verified the significant positive correlation between the three indicators, which will help us understand the typing system better.

Obviously, the system should not always output the index with the highest score when typing digits and punctuation marks, during which the statistical decoder is not applied. Significantly low matching scores even at Rank 1 are likely to result in false matches, and users need at least two more editing commands to correct the possible mistakes. We conducted an analysis to determine when to reject the recognition. As discussed in section 3.3, we will accept the recognition if the top score $s_1\,(I, G) > \mu,$ and $\frac{s_1(I,G)}{s_2(I,G)} > \lambda, (\lambda \geq 1)$. We analyzed the results of all users in the pilot study and calculated the expected risk of each typing time in Equation (3), as shown in Figure 8. When $\mu$ is greater than 0.06, the expected risk of one pressing time $R$ hardly changes with $\lambda$. Because $s_1$ is generally true matching score when it is large enough, and $s_2$ is usually much smaller at this point, which will however lead to rather high false rejection. The time cost will reach the minimum value when $\lambda$ is 1.152, and $\mu$ is 0.0348. We finally chose the output condition as displayed in Equation (2). Based on the optimized rejection

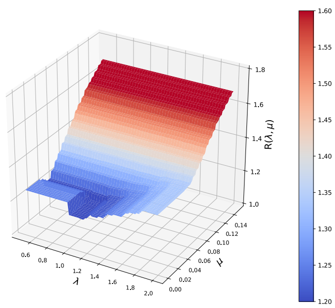


Fig. 8. 3D heat map of how choices of rejection affect typing efficiency. As the parameters increase, it's easier to reject recognition, which will lead to a higher false rejection rate. On the contrary, if the parameters are very low, the false recognition rate will increase. We found the parameters corresponding to the minimum expected risk of one pressing time by approximation methods.

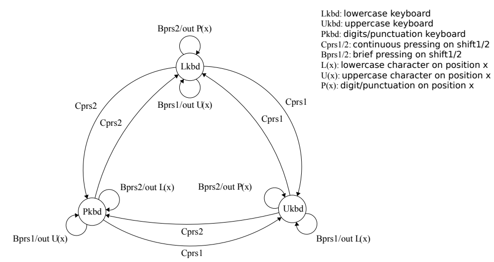



Fig. 9. The mechanism of keyboard state switching. There are two switching modes between the three keyboards, continuous pressing and brief pressing. If pressing shift continuously, the output will remain in the switched keyboard state. If pressing shift briefly, the keyboard will be in the new state for the next character, but then it goes back to the original keyboard state. The two modes are designed to adapt to different typing needs.

parameters, the false rejection rate of all 65 regions in the pilot study is 0.73%, and the mean rejection rate is 9.23%, slightly higher than mean recognition error rate of 8.78%.

When asked about their preferred typing layout, 75% of the participants chose the alphabetical layout, because they are more familiar with the alphabetical order. In fact, the experience for QWERTY layout cannot be easily transferred to virtual keyboards. Considering the user experience during the test, participants were asked to select 10-30 preferred regions, which will be used as an reference for layout designs. There is little difference in the region selection of users, which are highly consistent with the regions with high recognition rates.

## 4.2 Rich Typing Function Design

Besides basic letter typing, we also designed richer typing function which includes most of the function keys implemented on physical keyboards. Richer function includes space, backspace, enter, four arrow keys, two shift keys and continuous pressing function. Shift keys and continuous pressing function can provide more typing possibilities in limited input area, which will be described in details.

*4.2.1 Switching Keyboards Function.* In addition to the main keyboard, two switching keyboards, including uppercase keyboard and digits/punctuation keyboard, are enabled to expand the input symbol set. Figure 9 shows how the switching keyboards work. One brief pressing on shift keys will temporarily switch the keyboard, that is, it will return to the original keyboard after typing one character. Continuous pressing on shift keys will switch the keyboard permanently, and the subsequent typing will always apply to the switched keyboard, unless pressing shift keys continuously again to return to the original keyboard. The switching between the three keyboards and the design of two switching modes make it possible to type complex sentences including various symbols in a relatively simple way and a fast entry speed.

*4.2.2 Continuous Pressing Function.* Continuous pressing function is used for switching keyboards, switching characters, or typing the same character continually. It can effectively enrich the keyboard function and speed up the input speed. When pressed continuously, the system will only perform the recognition once. After one successful fingerprint recognition, a cyclical function is enabled to detect whether the finger exists, and the detecting time is negligible compared with fingerprint recognition time. If the pressing time is detected to exceed the threshold, the typing system will output the same character with reasonable intervals. Moreover, continuous pressing on the space key is used for selecting target words from the candidate set. Parameters like sleep time were reasonably adjusted during the test for the best user experience.

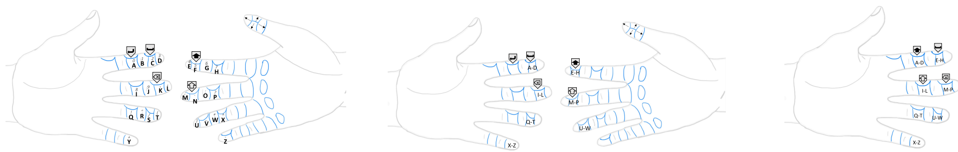
## 4.3 Design Space

There are plentiful possible layout designs by the arrangement and combination of keys and finger regions. Based on the pilot study discussed in section 4.1, we designed three typing strategies, which are divided into three interaction sub-spaces. The default layout in each of the three strategies can be modified by the users in real applications. The user needs to adjust the mapping between keys and finger regions in the central module and then the mapping is transmitted to the VR display module to update the display of layout. Therefore, PrinType can realize the mapping from any position to any character in theory and meet the needs of individuation. But usually it is not necessary to modify the default layouts. Moreover, unfixed layouts will make quantitative experiments difficult. The default layouts are mainly based on the recognition performance in the pilot study, while considering the symmetry and space utilization of the overall design.

*4.3.1 Hand-to-hand Interaction.* According to the recognition performance and practicality, we selected 26 symmetrical regions on both hands for letter entry in this strategy, and arranged them alphabetically, as shown in Figure 10(a). On the virtual keyboard shown in Figure 4, the second phalange, the DIP, the finger belly, and the fingertip are chosen as the regions for inputting letters. Except for the little fingers, the rest of the 6 fingers each corresponds to 8 letters. The backspace key and the space key lie on the sides of the two index fingers respectively, and the two shift keys on the sides of the two middle fingers. These four keys are placed in regions that are easy to press, because they will be used frequently. Layouts of the uppercase keyboard correspond to the main keyboard, while 10 digits and commonly used punctuation symbols are placed successively on the digits/punctuation keyboard. Furthermore, the region in the right thumb can be divided into four regions, corresponding to four arrow keys.

*4.3.2 Fingertip Interaction of Two Hands.* The second strategy considers the possible poor recognition rate of some users' joints and the inconvenient interaction between hands. We further simplified the layouts and reduced the complexity of interaction using a statistical language model. Compared to the first strategy, only 7 finger regions are required to type letters, that is, the keys of each finger are compressed into one, as shown in Figure 10(b). When typing words, the language model is enabled to predict each word based on the recognition sequence, and then users select the target word from the candidate words by pressing the space key continuously. When typing punctuation marks, the language model is not applied and the characters on the same position can be quickly switched and selected by pressing continuously. As described in 4.2.2, continuous pressing function only performs the fingerprint recognition once, and switching is performed by detecting the existence of the fingerprint pressing, so the time cost is negligible. We adjusted the threshold of the switching time to the most comfortable control state for the users. This strategy will significantly reduce the quantity of registered finger regions and improve the recognition rate.

*4.3.3 Finger Interaction on the Same-sided Hand.* The third strategy is basically the same as the second, but is optimized from the perspective of application scenarios. It is possible to type with one hand by placing all 7 finger regions of the letters in one hand, as illustrated in Figure 10(c). These regions include bellies of the fingers and the second phalanges. The shift keys, space and backspace will be placed on the four sides of the index and middle fingers.



(a) Strategy 1 (b) Strategy 2 (c) Strategy 3

Fig. 10. The default layouts in the three typing strategies. The mapping between digits/punctuation and letters in the latter two layouts is consistent with the first layout.

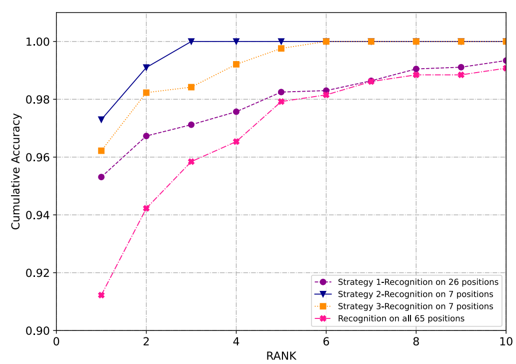
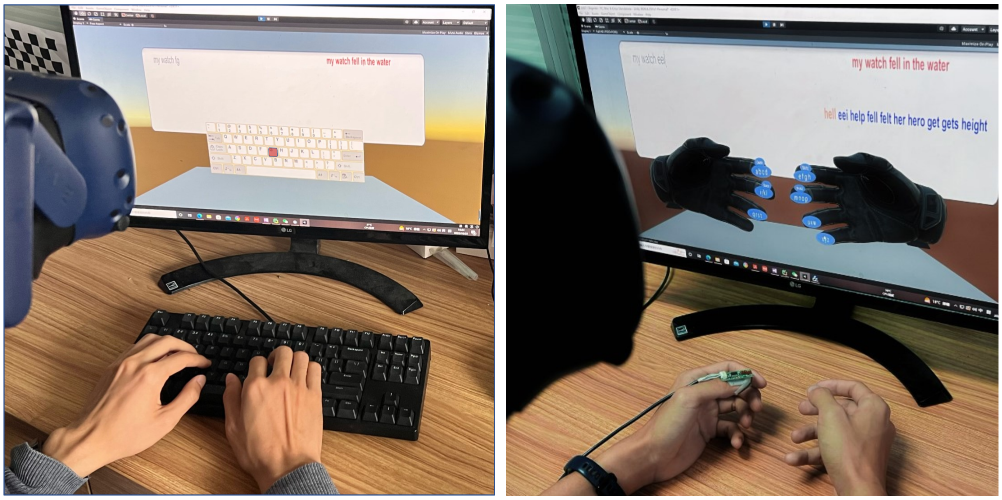



Fig. 11. CMC curves of different typing strategies. Recognition performance on 26 regions and 7 regions are much better than the recognition on all 65 finger regions, because the size of gallery is smaller and regions with bad recognition performance are excluded.

*4.3.4 Discussion.* We will analyze the above three interactive typing strategies from perspective of recognition performance. By comparing the CMC curves of the 26 and 7 regions in Figure 11, it can be seen that, at the expense of pressing time, Rank 1 recognition rate of 7 regions is higher without considering the recognition rejection. It can be inferred that the latter two strategies have fewer registration templates, and therefore have better recognition performance. Rank 1 recognition rate of both 7 and 26 regions testing is over 95%, which indicates good recognition performance in the small-scale gallery. It can be noticed that the variation between Rank 1 and Rank 2 is large in all CMC curves. Errors occur when users press the regions inaccurately, or the input fingerprints are of poor quality, which usually leads to low Rank 1 matching scores. In fact, when the recognition fails, the true matching score is usually close to Rank 1 matching score. Therefore, rejecting recognition by comparing the top two scores is an effective way to reduce the error rate. By setting thresholds of rejecting recognition, higher recognition rates will be achieved, especially for typing punctuation marks. Although this increases the frequency of key pressing, correcting the error itself will take more typing time due to pressing the backspace key.

## 5 SYSTEM EVALUATION

The goals of our controlled evaluation are: (1) to assess the overall typing performance of participants enabled by PrinType; (2) to observe the differences in typing performance of different participants in different input




(a) (b)

Fig. 12. The images show a user typing a simple sentence in VR with a physical keyboard and with PrinType (Strategy 2) respectively.

strategies and different humidity conditions; (3) to investigate the learnability of each input strategy and the preference of participants; and (4) to explore the empirical human performance potential of the particular text entry strategy with the micro metrics of performance in virtual reality.

## 5.1 Study Design

We recruited 12 participants (6 males, 6 females) between ages of 18 and 25. Participants were required to wear the Vive Pro HMD during the test and wear the fingerprint sensor on one's thumb which will record each pressing duration, interval duration, and the fingerprint images. The user study was a six-day longitudinal study, lasting about 40 minutes each day with three phases: preparation phase, testing phase, and feedback phase.

*5.1.1 Preparation Phase.* In the first day of preparation phase, participants were guided to familiarize themselves with the typing system, and they will follow the prompts to select the layout and register their fingerprints. Participants will be given 10 minutes to practice typing based on visual feedback from the HMD, during which they can adjust their layouts if they find particular regions are not well-recognized. We recorded the three layouts of each participant before testing.

*5.1.2 Testing Phase.* We used a within subjects 4×2 factorial design with two independent variables: input method (typing in VR with a physical keyboard, hand-to-hand interaction, finger interaction of two hands, and finger interaction within one hand) and input task (simple sentence and complex sentence). Furthermore, the learnability of an input method is reflected by observing how one learns the typing technique and how typing performance progressed as the testing days increase. The main purpose of the design is to compare the typing performance of the baseline method (physical keyboard) and three different strategies of PrinType in a fair way. The experiment setup of typing with a physical keyboard and with PrinType is shown in Figure 12. Participants could not see the keyboard or their hands directly in all four methods, and only static layouts were shown in the display during test to remind them of where the keys are. It is very likely that using a video passthrough or tracked keyboard and hands in VR [3] would improve the performance of the baseline method, from which PrinType would also benefit. To make the setup simple and fair, our experiments did not rely on video passthrough or keyboard and hand tracking.

Participants can see the prompt sentence they need to type in the HMD, and start or stop a timer for each entry by pressing the enter key. Participants were allowed to correct the current character by using the backspace key, and they were told in advance to balance typing accuracy and speed. Each participant needed to input a total of 4×12=48 simple sentences and 4×12=48 complex sentences with four different input methods each day. Simple sentences were selected from a public database [84], which do not contain uppercase letters, digits, or punctuation marks. A language model is used in the simple sentence test of PrinType, and participants can select the target word from the candidate words by pressing the space key continuously. Complex sentences were selected from a self-created database (see the supplemental material), and they involve lowercase and uppercase letters, digits, punctuation, special symbols, etc., so as to simulate rich text entry in VR office. The statistical decoder is not applied in the complex sentence test. It is worth noting that the typing test with a physical keyboard does not use a language model (following the conventions of previous studies in [49] and [21]).

PrinType will automatically record press duration, the captured fingerprint image, each matching result, each output character, and the times of deleting. These data will help us measure the typing performance of each user under different layouts of PrinType. The main evaluation metrics include Words Per Minute (WPM), Not Corrected Error Rate, and Corrected Error Rate.

The input speed of sentences is calculated as:

$$WPM = \frac{\left|\bar{L}\right| + 1}{\bar{T}} \times 60 \times \frac{1}{5} \quad , \tag{8}$$

where $\left|\bar{L}\right| + 1$ is the average length of characters in each input plus the enter character at the end, and $\bar{T}$ is the average typing time recorded in seconds in each input. The average length of the word in WPM is assumed to be 5 characters.

The error rates are defined based on character level without taking editing commands into account:

$$Not\ Corrected\ Error\ Rate = \frac{INF}{N} \quad , \tag{9}$$

$$Corrected\ Error\ Rate = \frac{IF}{N} \quad , \tag{10}$$

where $IF$ is the average number of characters typed incorrectly but fixed, $INF$ is the average number of substitutions, insertions or deletions required to align the observation sequence with the target sequence in each input, and $N$ is the average number of characters to be typed.

We test the robustness of recognition under wet and dry conditions in the first day of the user study. Since the image quality of fingerprints is easily affected by humidity conditions, we need to verify the reliability of the typing methods through controlled experiments. We sprayed water droplets on the hands of 12 participants to simulate high-humidity samples, and then we performed another complex sentence test of Strategy 1. Repeated recognition test was operated after generating dry fingerprints with a hair dryer. The results were compared with the full-keyboard recognition in the complex sentence test under normal condition.

*5.1.3 Feedback Phase.* Participants were asked to rate their dependence on the visual feedback from layouts under each strategy every day. Lower rating means they rely less on the feedback. And in the last day, participants rated each strategy with level of learnability, accuracy, speed, convenience, and overall preference on a scale of 1-5. Higher rating means a better experience.

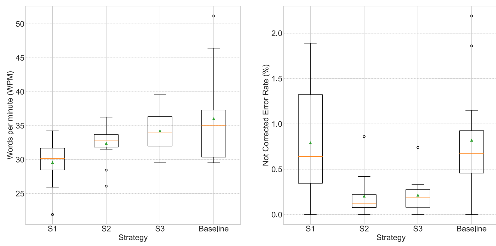



Fig. 13. Box plots of mean entry speed (WPM) and not corrected error rate (%) in typing simple sentences using different strategies.

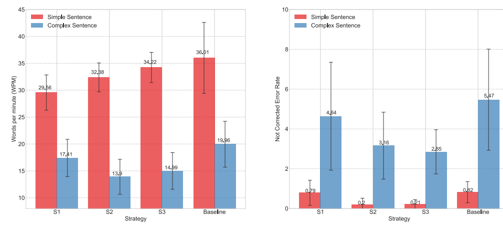



Fig. 14. Mean entry speed (WPM) and not corrected error rate (%) in typing simple sentences and complex sentences using three strategies of PrinType and the baseline method (black bars represent standard deviation).

## 5.2 Text Entry Results

In this section, we analyze the macroscopic typing performance in the baseline method and three typing strategies of PrinType. Different typing tasks are evaluated in terms of typing speed and error rate, and the robustness is tested under different humidity. We also summarize the rating under different subjective measures.

*5.2.1 Entry speed and error rate.* As shown in Figure 13, we compare the mean entry speed and not corrected error rate of simple sentences in different strategies. Note that the user had only 10 minutes of training before the test. All three strategies of PrinType achieve average entry speed of above 29 WPM and not corrected error rate of less than 0.8%, among which the first strategy has the worst performance, possibly due to participants' unfamiliarity with the layout with more keys and higher recognition error compared with Strategy 2 and 3. Typing in VR using the physical keyboard, no language model though, is the fastest among four strategies (similar to the results in [49] and [21]), but its not corrected error rate is the highest. Moreover, some participants have significantly faster typing speed or higher error rate in the baseline method. One important reason of high not corrected error rate is that users did not choose to correct these characters considering the balance between typing time and error rate. Therefore, not corrected error rate is not able to fully reveal the overall accuracy performance of the keyboards. Combining it with micro metrics, we will further explore the differences between keyboards and evaluate underlying user performance in the next section.

For comparison, PrinType's entry speed and error rate compares favourably to other thumb-to-finger interaction methods, as shown in Table 3, revealing that PrinType is easier to type compared to other similar typing methods. For a more comprehensive comparison, Table 3 also shows three key features of these typing techniques: use of statistical decoder, support of full character set, and real-time visual feedback of keystrokes. Methods like TipText [77] place multiple keys in a very small finger region, so a statistical decoder is a must to assist in typing accurately. In PrinType, it is possible to type without a statistical decoder (in the complex sentence test), and if the decoder is used, users can type each word faster by choosing the target word before typing the whole word. In addition, most of the other methods in Table 3 do not support full character set because the number of keys is greatly limited by the size of the sensor in thumb-to-finger interaction methods. It is also worth noting that the virtual keyboard of PrinType is displayed in VR as visual feedback because the symbol set is very large and the layouts of PrinType are different from the familiar QWERTY layout.

In terms of typing complex sentences, entry speed of all strategies decrease greatly, as shown in Figure 14. Complex sentences have special symbols, digits and uppercase letters, which require more pressing for one character, leading to longer typing time for each character compared with simple sentence test. In addition, the statistical decoder is not applied in the complex sentence test, which affects Strategy 2 and 3 most, because users

Table 3. Comparison of typing speed and error rate of different methods.

| | **WPM** | **Error Rate (%)** | **Employ Statistical Decoder** | **Support Full Character Set** | **Visual Feedback** |
|---|---|---|---|---|---|
| TipText [77] | 11.9 | 0.3 | ✓ | ✗ | ✗ |
| BiTipText [76] | 23.4 | **0.03** | ✓ | ✗ | ✗ |
| FingerText [40] | 27.9/30.9/30.4/33.1 | 4.10/2.87/1.44/0.79 | ✗ | ✗ | ✓ |
| FingerT9 [75] | 3.43 | 11.14 | ✓ | ✗ | ✓ |
| RotoSwype [27] | ~12 | ~1 | ✗ | ✗ | ✓ |
| HiFinger [35] | ~8 | ~5 | ✗ | ✓ | ✓ |
| Ours 1 | 29.56 | 0.79 | ✓ | ✓ | ✓ |
| Ours 2 | 32.38 | 0.20 | ✓ | ✓ | ✓ |
| Ours 3 | **34.22** | 0.21 | ✓ | ✓ | ✓ |

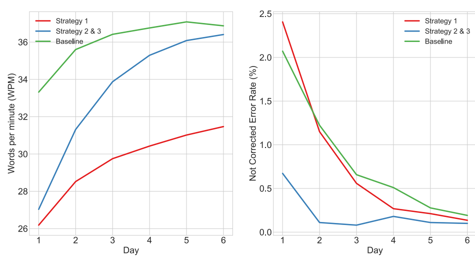



Fig. 15. The average input speed (WPM) and not corrected error rate (%) in the simple sentence test.

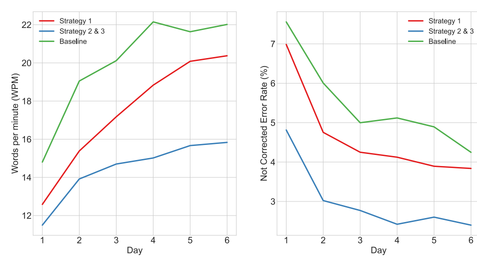



Fig. 16. The average input speed (WPM) and not corrected error rate (%) in the complex sentence test.

need to choose each character precisely by continuous pressing. Higher error rates in PrinType are mainly due to participants' unfamiliarity with using shift keys and the larger symbol set. For the baseline method, although we provide a QWERTY layout for users, it is not always easy to find the accurate positions of punctuation marks intuitively. In general, using the physical keyboard is faster in both typing tasks in VR, however, it is not accurate enough. Since other typing methods using thumb-to-finger interaction do not support full character set, there is no previously reported performance for comparison.

We summarized the average input speed and not corrected error rate in the simple and complex sentence test in Figures 15 and 16, where Strategy 2 and 3 are combined because their results are very close. Participants achieve entry speed of 31.47, 35.96, and 36.87 WPM with about 0.1% not corrected error rate in the last trial of the simple sentence test of PrinType. Strategies of PrinType show a continuous increasing input speed, which suggests that PrinType involves more finger interaction compared to typing with a physical keyboard, and users need to take more time to learn to type efficiently. Not corrected error rates decrease greatly on the second day in all test, because participants may re-register part of their finger regions or adjust their layouts after the first trial so that they ensure all the keys are well-recognized. At the end of the simple sentence test, not corrected error rates of all strategies are close, which indicates that after becoming familiar with the exact position of each key, participants have the same probability of refusing to correct errors. Based on this error rate, the highest entry speed of each participant reflects his/her potential typing speed. The current study shows that the entry

Table 4. Robustness test of typing system in different humidity conditions.

| | Error (%) | Reject (%) |
|---|---|---|
| **Normal Humidity** | 1.63 | 4.66 |
| **High Humidity** | 1.76 | 4.82 |
| **Low Humidity** | 1.66 | 4.71 |

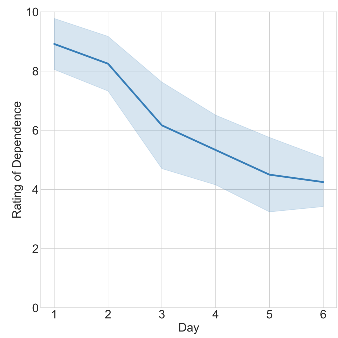



Fig. 17. The rating of dependence on the visual feedback of PrinType.

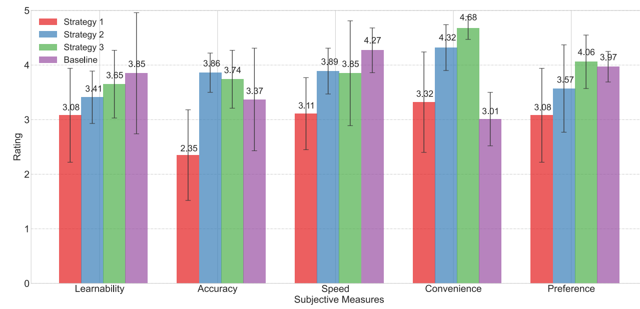



Fig. 18. Mean of subjective rating on learnability, accuracy, speed, convenience, and overall preference, using a scale of 1-5. Higher rating is better (bars represent standard deviation).

speed limit of PrinType is slower than the physical keyboard. However, thanks to better feedback of PrinType, which leverages innate proprioceptive ability of their own fingers, PrinType is superior to physical keyboards in accuracy in VR scenarios.

*5.2.2 Robustness.* Fingerprint recognition is susceptible to humidity of skin, so it is necessary to evaluate the robustness of typing under different humidity conditions. By creating dry and wet conditions, recognition performance in typing tests (no statistical decoder) is assessed. Table 4 shows the ratio of recognition error and rejection under different humidity conditions. The recognition error rate and rejection rate increase by 7.39% and 3.32% respectively in high humidity, 1.81% and 1.06% in low humidity. The test reveals that typing performance is hardly affected by low humidity, while high humidity has a slight effect on the results. Considering the extreme humidity we simulated, the robustness of the typing system will be guaranteed in normal weather conditions.

*5.2.3 Subjective Measures.* We can notice from Figure 17 that users generally became more familiar with PrinType and relied less on visual feedback from layouts with the testing days increasing. In the feedback phase, participants rated each strategy with different levels on a scale of 1-5, as shown in Figure 18. The ratings on accuracy and speed of the three strategies are consistent with the test results, that is, Strategy 2 has the highest accuracy rating and physical keyboards have the highest speed rating. The third strategy has the highest mean and low standard deviation in convenience and overall preference rating, suggesting that single-handed typing is favored by users in overall.

## 5.3 Micro Metrics

In this section, we analyze metrics including times of different steps and patterns of errors. These micro metrics will assist us in better understanding PrinType in three ways: (1) The comparison of metrics among different strategies indicates the strengths and weaknesses of each strategy, thus giving us directions to ameliorate the

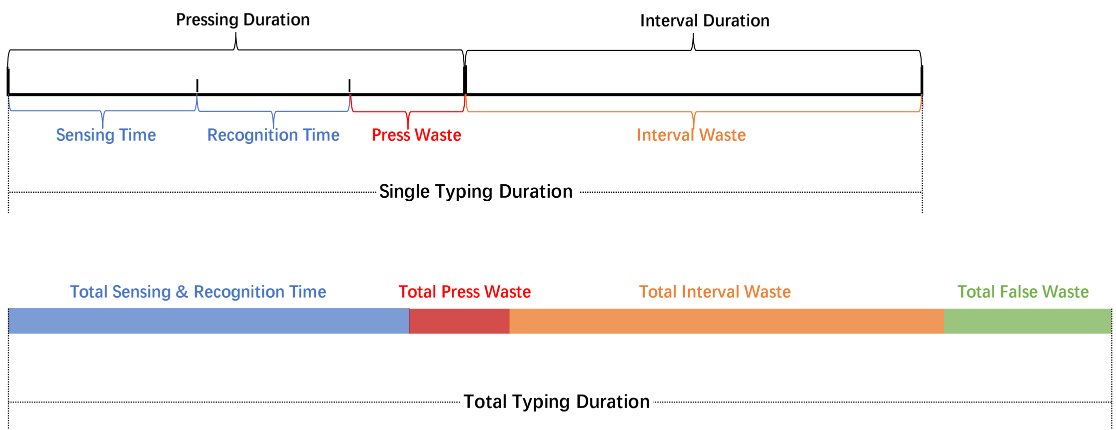



Fig. 19. Micro composition of single typing duration and total typing duration respectively.

Table 5. Time waste comparison across three strategies and participants.

| | | Mean Recognition Time (ms) | Mean Press Waste (ms) | Mean Interval Waste (ms) | Total False Waste (ms) |
|---|---|---|---|---|---|
| **Strategy 1** | All participants | 72.3 | 33.9 | 211.6 | 5253.7 |
| | Group A | 69.4 | 19.3 | 182.1 | 4621.6 |
| | Group B | 74.6 | 42.1 | 283.4 | 5321.6 |
| | Rate of Increase | 7.0% | 54.2% | 35.7% | 13.2% |
| **Strategy 2** | All participants | 23.5 | 86.9 | 242.5 | 7853.8 |
| | Group A | 22.6 | 62.8 | 163.8 | 7254.7 |
| | Group B | 23.9 | 118.3 | 294.1 | 9537.2 |
| | Rate of Increase | 5.4% | 46.9% | 44.3% | 23.9% |
| **Strategy 3** | All participants | 23.9 | 82.1 | 216.7 | 8368.3 |
| | Group A | 23.1 | 57.9 | 148.6 | 7574.2 |
| | Group B | 24.2 | 103.7 | 258.5 | 10035.8 |
| | Rate of Increase | 4.5% | 44.2% | 42.5% | 24.5% |

typing system. (2) We are enabled to measure users' typing habits and fingerprint quality, leading us to design a more suitable keyboard layout for each user. (3) By comparing performances among users, we can identify the key underlying metrics affecting typing efficiency, and guide users to make better use of PrinType. Furthermore, it is possible to explore the empirical human performance potential of a particular text entry strategy.

*5.3.1 Time Analysis.* We analyze the typing time under each strategy of PrinType for the complex sentence test, which does not include the language model. As shown in Figure 19, we delineate the micro composition of single typing duration and total typing duration respectively. The total typing duration is made up of pressing duration and interval duration for correct typing, and wasted time caused by incorrect or invalid input. We compare three time consuming variables of PrinType: Press Waste, Interval Waste and False Waste. Press Waste is the pressing

Table 6. Mean Press Waste and mean Interval Waste across three strategies in the last trial.

| | Mean Press Waste (ms) | Mean Interval Waste (ms) |
|---|---|---|
| **Strategy 1** | 23.6 | 173.7 |
| **Strategy 2** | 47.8 | 151.8 |
| **Strategy 3** | 43.0 | 139.4 |

duration minus sensing time and recognition time, Interval Waste is the inter-key interval duration, and False Waste is the difference between the total typing duration and typing duration for correct input.

We divide the participants into three groups: the three with the best typing performance are Group A, the three with the worst typing performance are Group B, and the rest are Group C. We compare various indicators between them, as shown in Table 5. In the table, rate of increase is the increase ratio of Group B to Group A. It can be seen that Interval Waste takes the most time in all strategies and all groups. Recognition time of the latter two strategies is significantly lower than that of the first strategy because of the smaller gallery, and an one-way ANOVA test shows no significant difference among groups ($F(2,9) = 2.153$ for Strategy 1, $F(2,9) = 1.932$ for Strategy 2, and $F(2,9) = 1.573$ for Strategy 3).

In all three strategies, Press Waste has the highest rate of increase. However, an one-way ANOVA shows that the group has the most significant effect on Interval Waste in all strategies ($F(2,9) = 44.211$, $p < 0.001$ for Strategy 1, $F(2,9) = 17.214$, $p < 0.005$ for Strategy 2, and $F(2,9) = 20.892$, $p < 0.001$ for Strategy 3). The difference in user performance is mainly due to the interval time in all strategies. Long inter-key interval duration may be due to the users' unfamiliarity with the keyboard layout.

In addition to the significant difference in interval time, the waste of time caused by false recognition is also an important factor among groups ($F(2,9) = 12.352$, $p < 0.005$ for Strategy 1, $F(2,9) = 11.463$, $p < 0.005$ for Strategy 2, and $F(2,9) = 14.785$, $p < 0.005$ for Strategy 3). Although the layouts with fewer keys improve the recognition rate and shorten the recognition time, some participants are not familiar with the mode of switching characters by pressing continuously in the complex sentence test. As a result, users may miss the best timing to release their fingers, or even result in incorrect output. Therefore, the corrected error rate of the latter two strategies is higher, which also explains why Press Waste and False Waste in the latter two strategies are significantly higher than Strategy 1.

As shown in Table 6, the mean Press Waste and the mean Interval Waste of three strategies in the last trial decrease significantly. The participants may even have situations where Press Waste is less than 0, because they have learned to quickly release their fingers at the beginning of sensing. When users are more familiar with each keyboard layout of PrinType, they usually do not wait for the recognition result to come out and can reach the next finger region during recognition. It can be summarized that participants are able to improve typing speed by reducing the pressing duration and the interval duration.

In order to study the influence of finger interaction on typing speed, we compare the interval time of Strategy 2 and Strategy 3 in the last trial. After using PrinType for 5 days, the users are familiar with the finger region of each letter and interval time is not interfered by factors such as fingerprint recognition. An one-way repeated measures ANOVA shows that typing strategies have a significant effect on interval time ($F(1,11) = 10.318$, $p < 0.01$), which proves that if users are familiar enough with PrinType, one-handed typing (Strategy 3) is faster than typing with both hands (Strategy 2) because of more efficient finger interaction.

*5.3.2 Error Analysis.* The mean corrected error rates under the three strategies of the complex sentence test are 3.104%, 4.054%, and 4.216% respectively, while the corrected error rate reaches 28.385% in the baseline method. Although a QWERTY layout is displayed when typing with a physical keyboard, users are still prone to mistype

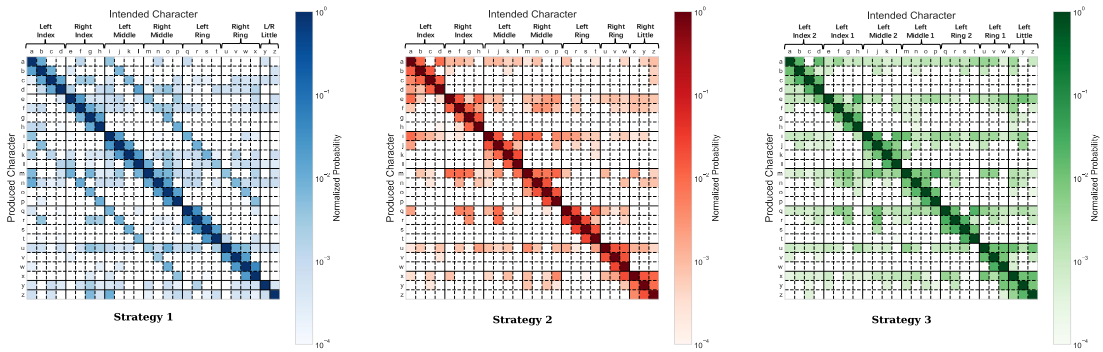
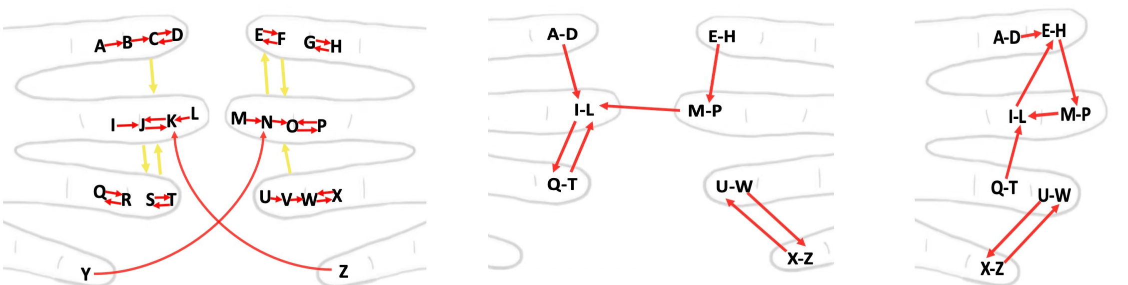



Fig. 20. Confusion matrixes showing the probability of produced vs intended characters under the three strategies. Each plot is ordered by the placement of keys along the finger so that the relationship between the most confused characters and their spatial positions can be easily found. The scale is set as logarithmic to better highlight the confusions, otherwise the errors are hard to discern due to very low error rates.




(a) Strategy 1 (b) Strategy 2 (c) Strategy 3

Fig. 21. Visualization of mistyping patterns with the highest error probability on each key (red arrows) or each finger (yellow arrows in Strategy 1). Among them, the typing error of letter y and z in Strategy 1 is mainly caused by recognition error, so the highest error rates of them are not high and there is no obvious pattern among different participants.

in the complex sentence test, which leads to slow entry speed. In PrinType, higher corrected error rate will directly lead to more False Waste time. Although Strategy 1 has the highest uncorrected error rate (0.79% in Table 3), its corrected error rate is the lowest, which indicates that entry errors of Strategy 1 are usually due to failed recognition of particular finger regions and users usually have to give up correcting them. Better recognition performance of the latter two strategies can ensure that errors are corrected as much as possible, which leads to lower uncorrected error rate but slower typing speed.

Confusion matrixes in Figure 20 visualize the pattern of mistypes under the three strategies. Horizontal and vertical thick lines are drawn to divide different fingers (Strategies 1 and 2) or different finger segments (Strategy 3). Since the plots are ordered according to the layouts, the most confused characters gather closer to the diagonal in all three strategies. Specifically, mistyping in Strategy 1 is usually caused by false pressing between adjacent keys on the same finger (a⟶b, b⟶a/c, c⟶b/d, d⟶c, etc.), as shown in Figure 21(a), while the errors in the latter two strategies are mostly from bad timing of releasing fingers during continuously pressing (a→b, b→a/c, c→b/d, d→c/a, etc.).

Furthermore, two regularly arranged diagonal lines can be seen in the confusion matrix of Strategy 1 in Figure 20 (i→a/q, j→b/r, k→c/s, etc.), which is mainly because the symmetrical layout of Strategy 1 results in users having fixed and similar mistyping patterns (right segments but wrong fingers), as shown in Figure 21(a). In general, pressing errors in Strategy 1 are usually presented as regular diagonal lines in the confusion matrix, while recognition errors have no obvious pattern. Nevertheless, the regions with higher recognition errors in Strategy 1 (i, j, q, r, w, etc.) are consistent with the results in Figure 4. High probability of errors often appear beneath the horizontal thick lines in Strategy 2 and 3 (i→a/e/m/q/u/x, etc.), which is due to false finger recognition or users pressing the wrong key, as shown in Figures 21(b) and 21(c).

Most of the errors mentioned above are auto-corrected by the statistical decoder in the simple sentence test. With matching scores as the input, the decoder will auto-correct the mistyped letter by predicting the input sequence as a reasonable word in the dictionary. It is possible to further improve the typing performance by introducing the above error patterns related to the keyboard layout into the statistical decoder.

## 6 DISCUSSION, LIMITATIONS, AND FUTURE WORK

### 6.1 Discussion

Text entry in virtual reality presents three major challenges: (1) Studies have shown that typists need to look down at the keyboard to verify their hand position, regardless of their degree of familiarity with the keyboard [64]. Since the hands are invisible to users wearing HMDs, it is almost impossible for them to input text as fast and accurate as usual [12]. (2) There are usually rich interactions in VR with different typing requirements. How to maintain a free and convenient user interaction experience while typing efficiently is a major challenge. (3) On-body virtual keyboards usually have limited input space constrained by the size of sensor. How to map the discrete high-density keys to the hands within limited area is a topic worth studying.

In terms of the first challenge, PrinType offers real-time visual feedback in virtual reality when touching fingers that serve as the virtual keyboard, eliminating the need for users to memorize complex keyboard layouts. In addition, research showed that people can accurately touch their finger segments with their thumbs, and thumb-to-finger interfaces support effective eyes-free interaction [34]. Eyes-free interfaces located on skin outperform interfaces on physical devices, which suggests that palm-based imaginary interfaces may have benefits for visually impaired users, potentially outperforming the touchscreen-based devices they use today [28]. In PrinType, all three strategies fully leverage users' innate proprioceptive ability of their own fingers [73, 74], and users can theoretically type without any visual feedback.

Secondly, the three strategies are suitable for people with different habits in different scenarios. For example, the first strategy is similar to full keyboard typing, while the latter two strategies are enlightened by nine-key typing. The layout of the first strategy is more intuitive, and its larger input space will provide richer typing function. The third strategy, which achieves same-sided hand (SSH) typing, has greater application value in other scenarios. SSH interaction can offer benefits by freeing the other hand for tasks like carrying a bag. PrinType makes full use of the uniqueness and accuracy of the patterns on fingers, practically independent of the external environment and the positions of other mobile devices, enabling a private keyboard space. Moreover, the keyboards can be flexibly customized in PrinType when certain fingers are injured or hard to recognize, without the need to redesign the keyboard layout completely or to modify hardware devices, making it a flexible and user-friendly typing interaction.

Thirdly, PrinType first leverages fingerprint recognition technique to realize an effective, flexible, and fully functional interactive typing system. Thanks to the robust pattern matching algorithm, almost all the touchable finger regions can be endowed with symbols or functions. Key sharing can be conveniently realized by shift keys to further expand the symbol set and thereby enrich the possible functionality of the keyboard.

In addition to applications in VR/AR, our typing method has potential in smartwatches. PrinType does not rely on the touch of the screen, saving the valuable screen real estate. Furthermore, besides being used for typing, different functions can be assigned to different regions of fingers to enable a variety of applications like drawing and gaming.

### 6.2 Limitations and Future Work

While results from our evaluation show the usability of PrinType, the current technique has several limitations:

- Registration in PrinType is unavoidable for a new user. Fortunately, the registration time is within 2 minutes in Strategy 1 (the most time consuming strategy) and it is performed only once. This also means that PrinType can only be used by registered users, which is good for continuous and seamless user authentication [48], but may be inconvenient in situations where the device needs to be shared with others. Fortunately, it can be used by multiple people (such as family members) as long as their fingerprints are all registered in PrinType. Since fingerprints of different people are recorded with person identity, this does not affect the performance of typing.
- We exploited a language model for disambiguation, which will predict the intended input and automatically correct the mistyping characters. By considering both language model and error patterns specific to PrinType, faster typing speed and lower error rate are achieved. However, the decoder we designed is not powerful enough to correct all kinds of mistyping. Specifically, we mainly solve the false pattern of substitutions, without taking other patterns into account and we have to inform participants of how to type efficiently in PrinType before testing.
- Fingerprint recognition is more computationally complex than physical keyboards or touch-based typing techniques. In the case of a large number of keys (Strategy 1), the time of fingerprint recognition cannot be ignored. The speed of fingerprint recognition is critical for users who tend to confirm the character being typed before typing the next character. It is worth further study to improve the efficiency of fingerprint feature extraction and recognition algorithm.
- Skin quality may change with weather and physical condition, which has an adverse effect on fingerprint recognition. Although recognition under different humidity conditions was tested in our study, other factors have not yet be considered. To improve the performance for fingerprints of low quality, template update [70] and more powerful fingerprint processing and matching algorithms using deep learning [14, 69] should be explored.
- So far, the participants in our user study are young students with good fingerprint quality and fast learning of new techniques. The fingerprint recognition performance of elderly people is usually worse, and their learning of new keyboards may be slower. Therefore, fingerprint typing technology suitable for the elderly is a topic worth studying in the future.
- Because participants are not familiar enough with keyboard layouts of PrinType, especially due to their large symbol set, the virtual keyboard is displayed in VR as visual feedback currently. The typing performance without keyboard feedback can be studied in the future when users are familiar with PrinType or only letters are considered.
- There is a wide range of layout design based on recognition of different regions of hands, and we have only explored a small part of it. Other keyboard layouts adopted in [39, 74] can be combined with PrinType seamlessly. Besides finger patterns, the patterns on palm can also be recognized based on fingerprint sensor and integrated in future keyboard design. Moreover, a bimanual text input method can be developed by attaching fingerprint sensors to both thumbs, as demonstrated in [76]. Although this will increase the

hardware cost and may cause inconvenience, it can further improve the typing efficiency and enrich the functions of PrinType by identifying multi-touch events. The keyboard design and user performance with the fingerprint sensor attached to other fingers besides the thumb are also worth studying.

- The hardware of PrinType is unshaped with a large fingerprint sensor (1 $cm^2$) and an exposed processing board. The current processing board is large because it is a complete fingerprint recognition system including sensor, storage, processing, and communication unit. In fact, we just need the sensor and communication module, which will be much smaller. In the future work, we plan to simplify the fingerprint sensor module (removing unnecessary parts) and test the performance of fingerprint sensor with a smaller area. Furthermore, flexible fingerprint sensor may become commercially available in the future which will be more suitable as wearable devices [72].
- Fingerprints are sensitive personal information because they are already widely used for identification. When PrinType is used for large-scale practical applications, the protection of fingerprint data is a problem that must be solved. The solution could be similar to the way smartphones protect fingerprint data [1].
- PrinType is currently only used for typing, but the technique can easily be extended to other applications, such as mapping different regions of fingers to different APPs, control keys for a specific APP, etc. A large number of APPs and control keys can be supported by switching the panel with shift keys. This could make computing systems more accessible for people with visual impairments [23].

## 7 CONCLUSION

While it is not a new concept of using hands as a virtual keyboard, in order for this concept to become a widely used typing method, techniques that balance performance, functionality, portability, and cost are required. We presented PrinType, which uses fingerprint recognition to enable intuitive and efficient text entry for virtual reality. We conducted a pilot study to test the performance of our fingerprint recognition technique in different regions of fingers. Three alternative typing strategies were designed and evaluated through a 12-participants user study to demonstrate the feasibility of PrinType. Our study showed that PrinType's three layouts all have fast entry speed of over 29 WPM on average and over 31 WPM in the last trial. In general, the third strategy is most favored by users for its convenience. We also analyzed micro metrics of PrinType across different strategies and participants, which point out potential problems in typing method of users and current technique.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for providing many helpful suggestions that improved the paper and the participants for their participation in our studies. This work was supported in part by the National Natural Science Foundation of China under Grant 61976121.

## REFERENCES

[1] Android. 2022. Trusty TEE. https://source.android.com/security/trusty.

[2] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. 2021. Bringing full-featured mobile phone interaction into virtual reality. *Computers & Graphics* 97 (2021), 42–53. https://doi.org/10.1016/j.cag.2021.04.004

[3] Harry Baker. 2022. Best Wireless Bluetooth Keyboards For Meta Quest 2. https://uploadvr.com/best-wireless-bluetooth-keyboards-for-meta-quest-2/

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110, 3 (2008), 346–359. https://doi.org/10.1016/j.cviu.2007.09.014

[5] Sabah Boustila, Thomas Guégan, Kazuki Takashima, and Yoshifumi Kitamura. 2019. Text Typing in VR Using Smartphones Touchscreen and HMD. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 860–861. https://doi.org/10.1109/VR.2019.8798238

[6] Sidney Bovet, Aidan Kehoe, Katie Crowley, Noirin Curran, Mario Gutierrez, Mathieu Meisser, Damien O Sullivan, and Thomas Rouvinez. 2018. Using Traditional Keyboards in VR: SteamVR Developer Kit and Pilot Game User Study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*. 1–9. https://doi.org/10.1109/GEM.2018.8516449

[7] D. A. Bowman, C. A. Wingrave, J. M. Campbell, V. Q. Ly, and C. J. Rhoton. 2002. Novel Uses of Pinch Gloves for Virtual Environment Interaction Techniques. *Virtual Reality* 6, 3 (2002), 122–129. doi:https://doi.org/10.1007/s100550200013

[8] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. 2010. Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (2010), 2128–2141. https://doi.org/10.1109/TPAMI.2010.52

[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)*. ACM, 255–260. https://doi.org/10.1145/2501988.2502016

[10] Wenqiang Chen, Lin Chen, Meiyi Ma, Farshid Salemi Parizi, Shwetak Patel, and John Stankovic. 2021. ViFin: Harness Passive Vibration to Continuous Micro Finger Writing with a Commodity Smartwatch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 1, Article 45 (Mar 2021), 25 pages. https://doi.org/10.1145/3448119

[11] Samir Dash. 2017. BlueTap — The Ultimate Virtual-Reality (VR) Keyboard. https://medium.com/eunoia-i-o/bluetap-the-ultimate-virtual-reality-vr-keyboard-77f1e3d57d6f

[12] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *Human-Computer Interaction. Recognition and Interaction Technologies*, Masaaki Kurosu (Ed.). Springer International Publishing, Cham, 419–437.

[13] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance Envelopes of Virtual Keyboard Text Input Strategies in Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 289–300. https://doi.org/10.1109/ISMAR.2019.00027

[14] Joshua J. Engelsma, Kai Cao, and Anil K. Jain. 2021. Learning a Fixed-Length Fingerprint Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 6 (2021), 1981–1997. https://doi.org/10.1109/TPAMI.2019.2961349

[15] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. 2020. PinchType: Text Entry for Virtual and Augmented Reality Using Comfortable Thumb to Fingertip Pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. ACM, 1–7. https://doi.org/10.1145/3334480.3382888

[16] Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (Jun 1981), 381–395. https://doi.org/10.1145/358669.358692

[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)*. ACM, 81–82. https://doi.org/10.1145/259963.260056

[18] Ulrike Gollner, Tom Bieling, and Gesche Joost. 2012. Mobile Lorm Glove: Introducing a Communication Device for Deaf-Blind People. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (Kingston, Ontario, Canada) *(TEI '12)*. ACM, 127–130. https://doi.org/10.1145/2148131.2148159

[19] Gabriel González, José P. Molina, Arturo S. García, Diego Martínez, and Pascual González. 2009. Evaluation of Text Input Techniques in Immersive Virtual Environments, José A. Macías, Antoni Granollers Saltiveri, and Pedro M. Latorre (Eds.). *New Trends on Human-Computer Interaction: Research, Development, New Tools and Methods*, 109–118. https://doi.org/10.1007/978-1-84882-352-5_11

[20] Jens Grubert, Eyal Ofek, Michel Pahud, and Per Ola Kristensson. 2018. The Office of the Future: Virtual, Portable, and Global. *IEEE Comput. Graph. Appl.* 38, 6 (Nov 2018), 125–133. https://doi.org/10.1109/MCG.2018.2875609

[21] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Effects of Hand Representations for Typing in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 151–158. https://doi.org/10.1109/VR.2018.8446250

[22] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 159–166. https://doi.org/10.1109/VR.2018.8446059

[23] William Grussenmeyer and Eelke Folmer. 2017. Accessible Touchscreen Technology for People with Visual Impairments: A Survey. *ACM Trans. Access. Comput.* 9, 2, Article 6 (jan 2017), 31 pages. https://doi.org/10.1145/3022701

[24] 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

[25] Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. ACM, 49–60. https://doi.org/10.1145/2984511.2984576

[26] Jeremy Gummeson, Bodhi Priyantha, and Jie Liu. 2014. An Energy Harvesting Wearable Ring Platform for Gesture Input on Surfaces. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services* (Bretton Woods, New Hampshire, USA) *(MobiSys '14)*. ACM, 162–175. https://doi.org/10.1145/2594368.2594389

[27] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing Using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, 1–12. https://doi.org/10.1145/3290605.3300244

[28] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. 2013. Understanding Palm-Based Imaginary Interfaces: The Role of Visual and Tactile Cues When Browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, 889–898. https://doi.org/10.1145/2470654.2466114

[29] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. ACM, 441–450. https://doi.org/10.1145/2047196.2047255

[30] Zhenyi He, Christof Lutteroth, and Ken Perlin. 2022. TapGazer: Text Entry with Finger Tapping and Gaze-Directed Word Selection. In *CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. ACM, Article 337, 16 pages. https://doi.org/10.1145/3491102.3501838

[31] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, 1063–1072. https://doi.org/10.1145/2556288.2557130

[32] Adrian H. Hoppe, Leonard Otto, Florian van de Camp, Rainer Stiefelhagen, and Gabriel Unmüßig. 2018. qVRty: Virtual Keyboard with a Haptic, Real-World Representation. In *HCI International 2018 – Posters' Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 266–272.

[33] Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. 2016. Designing a Willing-to-Use-in-Public Hand Gestural Interaction Technique for Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, 4203–4215. https://doi.org/10.1145/2858036.2858436

[34] 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)*. ACM, 1526–1537. https://doi.org/10.1145/2858036.2858483

[35] Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, and Feng Chen. 2019. HiFinger: One-Handed Text Entry Technique for Virtual Environments Based on Touches between Fingers. *Sensors* 19, 14 (2019). https://doi.org/10.3390/s19143063

[36] Junhyeok Kim, William Delamare, and Pourang Irani. 2018. ThumbText: Text Entry for Wearable Devices Using a Miniature Ring. In *Proceedings of the 44th Graphics Interface Conference* (Toronto, Canada) *(GI '18)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 18–25. https://doi.org/10.20380/GI2018.04

[37] Youngwon R. Kim and Gerard J. Kim. 2016. HoVR-Type: Smartphone as a Typing Interface in VR Using Hovering. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (Munich, Germany) *(VRST '16)*. ACM, 333–334. https://doi.org/10.1145/2993369.2996330

[38] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, 1–9. https://doi.org/10.1145/3173574.3173919

[39] Falko Kuester, Michelle Chen, Mark E. Phair, and Carsten Mehring. 2005. Towards Keyboard Independent Touch Typing in VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Monterey, CA, USA) *(VRST '05)*. ACM, 86–95. https://doi.org/10.1145/1101616.1101635

[40] DoYoung Lee, Jiwan Kim, and Ian Oakley. 2021. FingerText: Exploring and Optimizing Performance for Wearable, Mobile and One-Handed Typing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. ACM, Article 283, 15 pages. https://doi.org/10.1145/3411764.3445106

[41] 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

[42] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2 (ICCV '99)*.

[43] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (Nov 2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[44] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. BlindType: Eyes-Free Text Entry on Handheld Touchpad by Leveraging Thumb's Muscle Memory. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 18 (Jun 2017), 24 pages. https://doi.org/10.1145/3090083

[45] Xinyao Ma, Zhaolin Yao, Yijun Wang, Weihua Pei, and Hongda Chen. 2018. Combining Brain-Computer Interface and Eye Tracking for High-Speed Text Entry in Virtual Reality. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. ACM, 263–267. https://doi.org/10.1145/3172944.3172988

[46] Ben Maman and Amit Bermano. 2022. TypeNet: Towards Camera Enabled Touch Typing on Flat Surfaces through Self-Refinement. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 567–576. https://doi.org/10.1109/WACV51458.2022.00064

[47] 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)*. ACM, 1073–1082. https:

//doi.org/10.1145/2556288.2556964

[48] Florian Mathis, John H. Williamson, Kami Vaniea, and Mohamed Khamis. 2021. Fast and Secure Authentication in Virtual Reality Using Coordinated 3D Manipulation and Pointing. *ACM Trans. Comput.-Hum. Interact.* 28, 1, Article 6 (jan 2021), 44 pages. https://doi.org/10.1145/3428121

[49] Mark McGill, Roderick Murray-Smith, Daniel Boland, and Stephen A. Brewster. 2015. A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. ACM, 177. https://doi.org/10.1145/2702123.2702382

[50] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 519–528. https://doi.org/10.1109/VR50410.2021.00076

[51] Kyungha Min. 2011. Text Input Tool for Immersive VR Based on 3×3 Screen Cells. In *Proceedings of the 5th International Conference on Convergence and Hybrid Information Technology* (Daejeon, Korea) *(ICHIT'11)*. Springer-Verlag, Berlin, Heidelberg, 778–786.

[52] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-Han Kim. 2015. TypingRing: A Wearable Ring Platform for Text Input. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (Florence, Italy) *(MobiSys '15)*. ACM, 227–239. https://doi.org/10.1145/2742647.2742665

[53] Peter Norvig. 2009. *Natural Language Corpus Data*. 219–242.

[54] Eyal Ofek, Jens Grubert, Michel Pahud, Mark Phillips, and Per Ola Kristensson. 2020. Towards a Practical Virtual Office for Mobile Knowledge Workers. *New Future of Work 2020* August, 3-5 (2020). https://doi.org/10.48550/ARXIV.2009.02947

[55] Alexander Otte, Tim Menzner, Travis Gesslein, Philipp Gagel, Daniel Schneider, and Jens Grubert. 2019. Towards Utilizing Touch-sensitive Physical Keyboards for Text Entry in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 1729–1732. https://doi.org/10.1109/VR.2019.8797740

[56] Anna Peshock, Julia Duvall, and Lucy E. Dunne. 2014. Argot: A Wearable One-Handed Keyboard Glove. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers: Adjunct Program* (Seattle, Washington) *(ISWC '14 Adjunct)*. ACM, 87–92. https://doi.org/10.1145/2641248.2641266

[57] Duc-Minh Pham and Wolfgang Stuerzlinger. 2019. HawKEY: Efficient and Versatile Text Entry for Virtual Reality. In *25th ACM Symposium on Virtual Reality Software and Technology* (Parramatta, NSW, Australia) *(VRST '19)*. ACM, Article 21, 11 pages. https://doi.org/10.1145/3359996.3364265

[58] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: An Eyes-Free VR/AR Symbolic Input Device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (Edinburgh, Scotland) *(VRST '14)*. ACM, 9–18. https://doi.org/10.1145/2671015.2671029

[59] Vaughan Pratt. 1998. Thumbcode: A Device-Independent Digital Sign Language. In *Finger-Joint Gesture Wearable Keypad, in Second Workshop on Human Computer Interaction with Mobile Devices*.

[60] VIVE Pro. 2018. The Professional-grade VR Headset. https://www.vive.com/us/product/vive-pro

[61] Vijay Rajanna and John Paulin Hansen. 2018. Gaze Typing in Virtual Reality: Impact of Keyboard Design, Selection Method, and Motion. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. ACM, Article 15, 10 pages. https://doi.org/10.1145/3204493.3204541

[62] R. Rosenberg and M. Slater. 1999. The Chording Glove: A Glove-Based Text Input Device. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29, 2 (May 1999), 186–191. https://doi.org/10.1109/5326.760563

[63] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 159 (Jan 2018), 23 pages. https://doi.org/10.1145/3161187

[64] TA Salthouse. 1984. Effects of age and skill in typing. *Journal of Experimental Psychology. General* 113, 3 (September 1984), 345—371. https://doi.org/10.1037//0096-3445.113.3.345

[65] Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. *LuminosoInsight/wordfreq: v2.2*. https://doi.org/10.5281/zenodo.1443582

[66] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-Based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 1–13. https://doi.org/10.1145/3173574.3174221

[67] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, 3643–3652. https://doi.org/10.1145/2702123.2702136

[68] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-Finger Text Entry on Everyday Surfaces via Bayesian Inference. In *CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. ACM, Article 497, 16 pages. https://doi.org/10.1145/3491102.3501878

[69] Yao Tang, Fei Gao, Jufu Feng, and Yuhang Liu. 2017. FingerNet: An Unified Deep Network for Fingerprint Minutiae Extraction. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (Denver, CO, USA). IEEE, 108–116. https://doi.org/10.1109/BTAS.2017.8272688

[70] Umut Uludag, Arun Ross, and Anil Jain. 2004. Biometric template selection and update: a case study in fingerprints. *Pattern Recognition* 37, 7 (2004), 1533–1542. https://doi.org/10.1016/j.patcog.2003.11.012

[71] James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. 2017. Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, 5457–5461. https://doi.org/10.1145/3025453.3025783

[72] An Byeong Wan, Heo Sanghyun, Ji Sangyoon, Bien Franklin, and Park Jang-Ung. 2018. Transparent and flexible fingerprint sensor array with multiplexed detection of tactile pressure and skin temperature. *Nature Communications* 9, 1 (2018), Article 2458. https://doi.org/10.1038/s41467-018-04906-1

[73] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y. Chen. 2015. PalmType: Using Palms as Keyboards for Smart Glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. ACM, 153–160. https://doi.org/10.1145/2785830.2785886

[74] 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 (Sep 2017), 21 pages. https://doi.org/10.1145/3130978

[75] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9: Leveraging Thumb-to-Finger Interaction for Same-Side-Hand Text Entry on Smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, 1–10. https://doi.org/10.1145/3173574.3173752

[76] 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*. ACM, 1–13. https://doi.org/10.1145/3313831.3376306

[77] 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)*. ACM, 883–899. https://doi.org/10.1145/3332165.3347865

[78] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. ACM, 539–548. https://doi.org/10.1145/2807442.2807504

[79] 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)*. ACM, 4479–4488. https://doi.org/10.1145/3025453.3025964

[80] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. 2018. PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics* 24, 11 (2018), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581

[81] Shumin Zhai and Per Ola Kristensson. 2012. The Word-Gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sep 2012), 91–101. https://doi.org/10.1145/2330667.2330689

[82] 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 (Sep 2017), 19 pages. https://doi.org/10.1145/3130985

[83] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, Omer Inan, and Gregory D. Abowd. 2018. FingerPing: Recognizing Fine-Grained Hand Poses Using Active Acoustic On-Body Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 1–10. https://doi.org/10.1145/3173574.3174011

[84] Mingrui Ray Zhang and Jacob O. Wobbrock. 2019. Beyond the Input Stream: Making Text Entry Evaluations More Flexible with Transcription Sequences. 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, 831–842. https://doi.org/10.1145/3332165.3347922

[85] Mingrui Ray Zhang, Shumin Zhai, and Jacob O. Wobbrock. 2022. TypeAnywhere: A QWERTY-Based Text Entry Solution for Ubiquitous Computing. In *CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. ACM, Article 339, 16 pages. https://doi.org/10.1145/3491102.3517686


## A STATISTICAL LANGUAGE DECODER

Suppose that users press on the regions of fingers $\vec{z}_l = [z_1, z_2, ..., z_l]$ to type the character sequence $\vec{y}_l = [y_1, y_2, ..., y_l]$. We set $\mathbb{Z}$ as the set of all used finger regions, and $z_{y_i}$ is the region of character $y_i$, $\mathbb{Y}_{z_i}$ is the set of characters of region $z_i$. After typing the character sequence, the recognition algorithm outputs a sequence of matching scores $\vec{X}_l = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_l]$. $\vec{x}_i$, with the length of $|\mathbb{Z}|$, consists of the unnormalized matching scores of the $i$-th input and all used finger regions. Then the purpose of the decoder is to estimate the most likely hidden state sequence $\hat{\vec{y}}_l$ from the observed sequence $\vec{X}_l$:

$$\hat{\vec{y}}_l = \arg\max_{\vec{y}_l} P(\vec{y}_l|\vec{X}_l) = \arg\max_{\vec{y}_l} P(\vec{y}_l, \vec{X}_l). \tag{11}$$

Based on the assumption of the hidden Markov model, Equation (11) can be rewritten as:

$$\begin{aligned}
\hat{\vec{y}}_l &= \arg\max_{\vec{y}_l} P(\vec{y}_l, \vec{X}_l) \\
&= \arg\max_{\vec{y}_l} \prod_{i=1}^{l} P(y_i|\vec{y}_{i-1}) P(\vec{x}_i|\vec{x}_{i-1}, \vec{y}_i) \\
&= \arg\max_{\vec{y}_l} \prod_{i=1}^{l} P(y_i|\vec{y}_{i-1}) P(\vec{x}_i|y_i) \\
&= \arg\max_{\vec{y}_l} \prod_{i=1}^{l} P(y_i|\vec{y}_{i-1}) \sum_{z_i \in \mathbb{Z}} P(\vec{x}_i|z_i) P(z_i|y_i) \\
&= \arg\max_{\vec{y}_l} \prod_{i=1}^{l} P(y_i|\vec{y}_{i-1}) \sum_{z_i \in \mathbb{Z}} \frac{P(z_i|\vec{x}_i)P(\vec{x}_i)}{P(z_i)} \frac{P(y_i|z_i)P(z_i)}{P(y_i)} \\
&= \arg\max_{\vec{y}_l} \prod_{i=1}^{l} \frac{P(y_i|\vec{y}_{i-1})}{P(y_i)} \sum_{z_i \in \mathbb{Z}} P(y_i|z_i) P(z_i|\vec{x}_i).
\end{aligned} \tag{12}$$

Consider that

$$P(y_i|z_i) = \begin{cases} \frac{1}{|\mathbb{Y}_{z_i}|}, & z_i = z_{y_i} \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

so

$$\hat{\vec{y}}_l = \arg\max_{\vec{y}_l} \prod_{i=1}^{l} \frac{P(y_i|\vec{y}_{i-1})}{P(y_i)} P(z_{y_i}|\vec{x}_i) \frac{1}{|\mathbb{Y}_{z_{y_i}}|}. \tag{14}$$

In Equation (14), $|\mathbb{Y}_{z_{y_i}}|$ is 1 for the full keyboard, and is 3 or 4 for the keyboards of Strategy 2 and 3. We used an $n$-gram model to estimate $P(y_i|\vec{y}_{i-1})$, a unigram model to estimate $P(y_i)$, and normalized matching scores to estimate $P(z_{y_i}|\vec{x}_i)$:

$$P(z_{y_i}|\vec{x}_i) = \frac{\exp(\tau\vec{x}_i)}{\sum \exp(\tau\vec{x}_i)}. \tag{15}$$

Then we applied a beam search to find the top $k$ most likely sequences according to the sum of the log probabilities and we further expended them based on the dictionary.

The parameter $\tau$ in Equation (15) is set to be 100 to smooth the probability and the parameter $n$ in the model is 7. The training dataset is derived from 1/3 million most frequent English words collected by Peter Norvig [53]. The dictionary consists of 12,131 English words, which are selected from [53] based on a frequency test [65]. In order to support typing out-of-vocabulary words, the uncorrected sequence is inserted in the second place of the candidate list. These sequences are displayed on Canvas in real time as candidate words for users to select.