DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays

ERIC WHITMIRE, MOHIT JAIN, DIVYE JAIN, GREG NELSON, RAVI KARKAR, and SHWETAK PATEL, University of Washington
MAYANK GOEL, Carnegie Mellon University

Input is a significant problem for wearable systems, particularly for head mounted virtual and augmented reality displays. Existing input techniques either lack expressive power or may not be socially acceptable. As an alternative, thumb-to-finger touches present a promising input mechanism that is subtle yet capable of complex interactions. We present *DigiTouch*, a reconfigurable glove-based input device that enables thumb-to-finger touch interaction by sensing continuous touch position and pressure. Our novel sensing technique improves the reliability of continuous touch tracking and estimating pressure on resistive fabric interfaces. We demonstrate DigiTouch's utility by enabling a set of easily reachable and reconfigurable widgets such as buttons and sliders. Since DigiTouch senses continuous touch position, widget layouts can be customized according to user preferences and application needs. As an example of a real-world application of this reconfigurable input device, we examine a split-QWERTY keyboard layout mapped to the user's fingers. We evaluate DigiTouch for text entry using a multi-session study. With our continuous sensing method, users reliably learned to type and achieved a mean typing speed of 16.0 words per minute at the end of ten 20-minute sessions, an improvement over similar wearable touch systems.

CCS Concepts: •**Human-centered computing →Interaction devices;** *Ubiquitous and mobile devices;*

Additional Key Words and Phrases: Wearable computing; head-mounted displays; thumb-to-finger; glove; eyes-free input

ACM Reference format:
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. *PACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (September 2017), 21 pages.
DOI: http://doi.org/10.1145/3130978

---

1 INTRODUCTION

Head-mounted displays (HMDs) for wearable virtual reality (VR) and augmented reality (AR) systems have seen a recent resurgence in interest and popularity. Fueled by advances in display and embedded technologies, head-mounted displays are poised to impact the way we work, play, and communicate. Currently, consumer applications of these technologies are focused on gaming and entertainment in stationary environments. Existing input techniques require socially awkward interactions or instrumented environments, limiting the broader usage of these devices in mobile settings. For example, head-mounted touch interfaces (Google Glass, Samsung Gear VR) or in-air gesture interfaces (Microsoft HoloLens) require raising a hand to eye level, which can be tiring and draw unwanted attention to the user. Speech input (Google Glass, Microsoft HoloLens) is useful for dictation and

---

Author's address: E. Whitmire, M. Jain, G. Nelson, R. Karkar, and S. Patel, Paul G. Allen School of Computer Science & Engineering, University of Washington; Mayank Goel, School of Computer Science, Carnegie Mellon University.

DOI: http://doi.org/10.1145/3130978

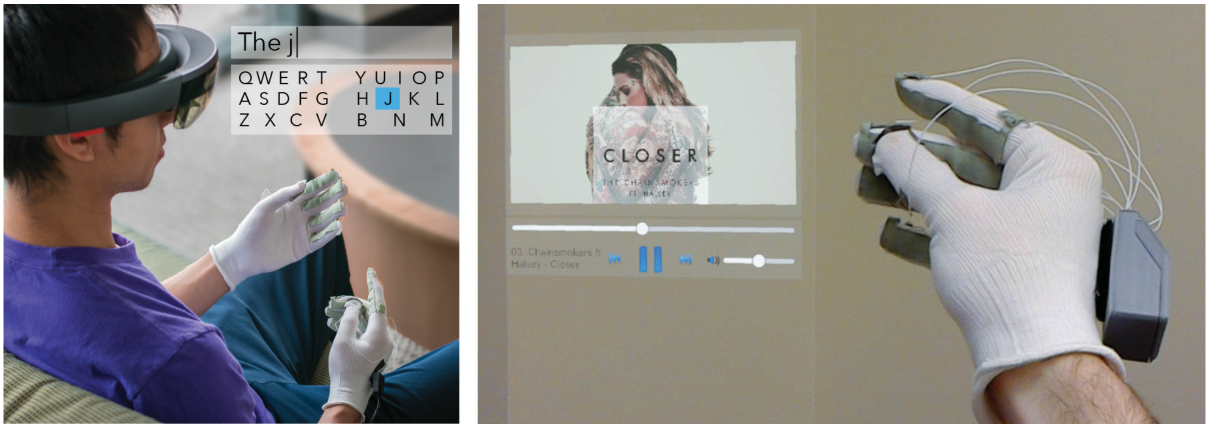


Fig. 1. DigiTouch is a touch-sensitive glove that explores the use of thumb-to-finger interaction for input and text entry. (left) Such an input technique is well-suited for use with virtual and augmented reality. (right) Controlling a HoloLens application using DigiTouch.

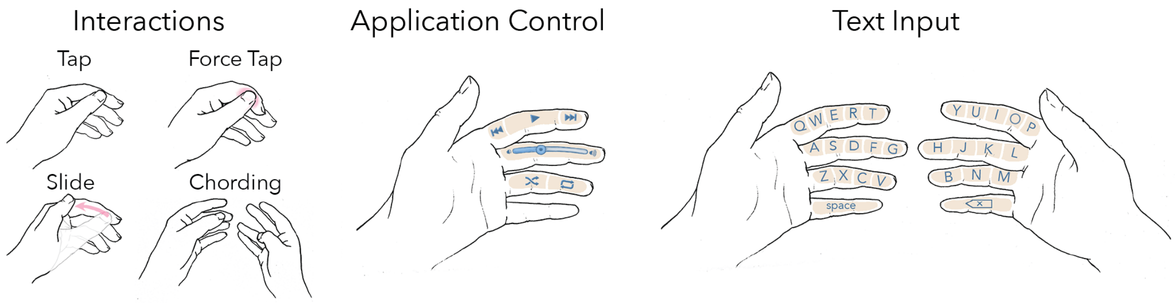


Fig. 2. (left) *DigiTouch* enables tapping, force tapping (tapping with pressure), swiping, and chording (two handed tapping) gestures. (center) Because DigiTouch senses continuous touch position, it enables an arbitrary configuration of input widgets, which can be customized for a particular application. (right) DigiTouch also enables text input by mapping a full QWERTY keyboard to the hands and reserving the pinky fingers for the space and backspace keys.

simple navigation, but may be disturbing to others and is not always socially acceptable [14, 33]. High-end VR devices (HTC Vive, Oculus Rift) use handheld positionally-tracked controllers for input. While these controllers offer immersive gaming experiences, they are not always appropriate for mobile input, as they require the user to hold an extra device. Therefore, **to enable broader use of head-mounted computing devices, there is an unmet need for input methods that are expressive, subtle, and portable.**

Thumb-to-finger interaction is a promising technique that can be performed discreetly, without large hand movements. Placing the input surface on the fingers enables fine-grained control that leverages both tactile and proprioceptive feedback. Furthermore, Huang et al. have shown that thumb-to-finger interactions are both comfortable and highly accurate [11]. Unlike many input methods that demand a particular posture during use, one can subtly swipe along a finger with the thumb with the arms at rest.

Though traditional optical hand trackers excel at hand pose detection and offer augmentation-free tracking, they do not provide enough granularity to precisely detect finger touch events and positions. Gloves, however, offer a number of advantages: freedom from occlusion and lighting problems may allow more subtle use and gloves may have fewer errors in recognizing input events. Gloves are particularly well-suited for input outside of traditional desktop computing environments (e.g. on the bus) or in situations where gloves are already commonly used (e.g. outdoors in cold weather). Because of the difficulty in creating non-contact haptics, people may also wear gloves when haptic feedback, such as force feedback or vibrotactile feedback, is desired. Though gloves might not be socially-acceptable in some situations, there is no one-size-fits-all input solution for head-mounted displays, and the ability to choose from a range of input devices for the situation will help make HMDs more ubiquitous.

In this paper, we present *DigiTouch*, a touch-sensitive glove that enables thumb-to-finger interaction for eyes-free input on wearable systems. DigiTouch uses thin, partially conductive fabric strips along the fingers and a conductive patch on the thumb pad (Figure 1). Each strip can sense the *continuous touch position* and *pressure* of the thumb as it touches the finger. This enables precise, yet subtle input through tapping, sliding, force-pressing, and two-handed chording gestures (Figure 2).

Unlike other data gloves [15, 16, 27, 32], which use only discrete touch regions, DigiTouch senses the continuous touch position of the thumb. This capability makes it reconfigurable; allowing it to be used for various tasks like target selection, slider control, and text entry. Depending on application requirements, different widgets of varying size can be mapped to different regions of a particular finger. Though others have demonstrated fabric-based touch interfaces that sense continuous input, such systems either require multiple layers of fabric [8] that hinder tactile feedback, or use sensor arrays [29] making them bulky and complex. It is also unclear how well these systems operate when bent or stretched, as doing so can change the electrical properties of fabrics. To overcome these challenges, we present a new technique for continuous sensing on fabric that uses only a single layer of fabric and a two-wire interface on each finger. DigiTouch accounts for the variable resistance as the fingers bend using current monitoring and time-multiplexed sensing.

DigiTouch is a general-purpose input device for AR/VR systems and can be used for different applications (Figure 2). For example, a user can dial a number using a ten-digit numeric keypad, move a virtual object by sliding along a finger, or control an application using any combination of buttons and sliders along the fingers. However, in evaluating DigiTouch, we decided to place special emphasis on text input using a split-QWERTY keyboard. The reason is three-fold: (1) Text entry is a challenging task in today's AR/VR systems and is a barrier to enabling more productive use-cases. (2) There is limited quantitative data on the evaluation of such wearable input systems for AR/VR applications, and text entry provides a well-established set of quantitative measures that helps in formalizing the system's performance. (3) The high density of keys using a full QWERTY keyboard in a fixed 2-dimensional space makes text entry a challenging task for the user, and a rigorous test for the usability and practicality of DigiTouch. We directly map a split-QWERTY keyboard layout to a user's fingers, as shown in Figure 2 (right). This closely resembles the two-thumb typing posture on a smartphone. From a longitudinal study with ten participants, we found that the participants quickly learned how to use DigiTouch for entering text. Their mean typing speed increased from 7.0 wpm (words per minute) to 16.0 wpm in 10 twenty-minute sessions. The participants also achieved a mean uncorrected error rate of 0.85% on the last session.

The main contributions of our work are:

(1) A reconfigurable touch-sensitive glove that senses continuous touch position and pressure, enabling thumb-to-finger interactions for wearable computing.
(2) A text entry system using thumb-to-finger interactions based on a split-QWERTY keyboard.
(3) A quantitative evaluation of the text entry capabilities of DigiTouch using a ten-session study with 10 participants.

2 RELATED WORK

2.1 Thumb-to-finger input

Due to the anatomy of the hand, touching the thumb to the fingers is a natural and expressive interaction. This interaction benefits from both tactile and proprioceptive (an innate sense of the body's position and movement) feedback. Prior work [11, 26] has explored this style of thumb-to-finger interaction for various applications. For example, *DigiTap* [26] uses a wrist-mounted accelerometer and camera to detect thumb-to-finger taps. The accelerometer detects when a tap occurs, and awakens the camera to observe where the tap occurred. It can identify discrete taps on 12 locations (three regions per finger). Our system senses continuous touch position, enabling reconfigurable input and an arbitrary number of touch regions per finger. The *DigitSpace* [11] prototype detects the thumb position along the length of a finger using a chain of Hall effect sensors. Though the technique is promising, their prototype is limited to two fingers and does not consider complex tasks such as text entry. Moreover, it requires instrumenting the fingers with rigid electronics, which could limit range of motion. Other projects explore subsets of thumb-to-finger input, e.g. *NailO* [12] and *Ringteraction* [6], which place small sensors on the finger to enable thumb gestures. In these systems, the interaction surface is limited to a small portion of a single finger. Saponas et al. [28] demonstrated a forearm-based electromyography device that can classify a set of hand gestures, including thumb-to-finger gestures. However, the discrete nature of the classification makes it unsuitable for fine-grained thumb-to-finger sensing.

To estimate thumb and finger positions, researchers have also explored vision-based techniques [1, 13] and techniques using magnets and magnetic transducers attached to the fingertips [2–4, 11]. *CyclopsRing* [1] uses a unique fisheye camera placed between the index and middle finger, to distinguish between a number of hand gestures, including several thumb-to-finger touches. However, it is unclear how accurately it can detect touch events and estimate the position of the thumb along the finger, due to occlusion issues. *Digits* [13] uses a wrist-mounted infrared camera to reconstruct a 3D model of the hand and fingers. However, it is designed for gesture-based input rather than continuous input. The magnetic approaches enable reliable continuous positional tracking; however, it is again difficult to detect contact between the thumb and finger. *FingerPad* [2] addresses this using an accelerometer affixed to the index finger to detect the contact impact. Though vision and magnetic tracking techniques may be viable hand-tracking solutions in the long term, they do not excel at detecting contact events. We chose to use a glove for our implementation to prioritize reliable touch detection.

2.2 Glove-based input

There are several examples of glove-based interfaces that have been proposed for input and text entry. Miller *et al.* [22] created a glove with a 2D input surface along the length of the fingers using an array of conductive threads, but only demonstrated the ability to perform simple targeting tasks. Though not a full glove, *Plex* [38] is a wearable finger covering that uses piezoresistive fabric to enable thumb-to-finger touches. Similarly, *TIMMi* [37] extends this to enable the reconstruction of finger bend and touch pressure. These techniques are similar to DigiTouch but have a limited interaction surface and do not explore interactions on all fingers. Moreover, these devices only support discrete touch points.

A few commercial data gloves[1] attempt full hand pose reconstruction using bend sensors in the finger joints. Theoretically, the necessary thumb and finger positions can be extracted from this data, but they are not accurate enough to reliably detect taps or estimate the relative position of the thumb and finger. Other commercial gloves, such as Peregrine Glove[2] uses a set of touch sensitive regions along the finger to lay out arbitrary buttons. However, they suffer from lack of tactile feedback due to the thickness of the glove and the touch sensitive

---

[1] CyberGlove: http://www.cyberglovesystems.com/, 5DT data glove: http://www.5dt.com/

[2] Peregrine Glove: http://theperegrine.com/

coil. DigiTouch uses a thin fabric layer and simple wiring that eliminates the bulk of many other glove-based approaches.

KITTY [15] is a glove that combines four contacts per thumb with six contacts on four fingers, to offer 48 button combinations, suitable for text entry. This results in an expressive, but complex set of possible touch points. By placing a continuous input space along each finger, we enable a split-QWERTY keyboard with a familiar layout. *Argot* [24] is a one-handed glove with 15 buttons, enabling text-entry using multi-tap and a T9 dictionary. Rosenberg and Slater [27] proposed a chording-based glove with seven buttons. With training, chording-based techniques can achieve high text entry speed (up 16.8 wpm after 10 hours [27]), but are difficult to master. With DigiTouch, participants were able to achieve similar typing speeds after just 3 hours of practice and did not have to learn a new key mapping. Our continuous sensing also enables an adjustable number and layout of buttons.

2.3 Other Text Entry Methods

Other systems for eyes-free text entry for wearables systems include vision-based approaches [21, 31, 34] and approaches using external devices [7, 17, 23]. Sridhar et al. [31], *Vulture* [21], and *PalmType* [34] use vision-based hand-tracking for text entry. Sridhar et al. use a mid-air chording technique and achieve 22.2 wpm, with participants entering a word until they reach their peak performance. *Vulture* is a mid-air word gesture keyboard that achieved 20.6 wpm at the end of 10-sessions, while *PalmType* uses the index finger of one hand to type on the palm of other hand, and achieved 4.6 wpm. Vulture and PalmType were prototyped using the Vicon motion tracking system, as they require fine-grained hand-tracking. These systems suffer from occlusion issues and in-air typing results in fatigue after extended use. *TiltType* [23] uses controlled tilting of the wrist for input, which can be fatiguing for the user, and *TypingOnGlass* [7] uses swiping on the frame of a Google Glass for text entry (8.7 wpm), which attracts attention and is socially awkward [33]. Twiddler[3] is a hand-held device for text entry using multifinger chording. Prior work has shown participants achieve 26.2 wpm after 400 minutes of practice. Though this outperforms most other systems for wearable computing text entry, DigiTouch enables additional interactions, like swiping, and does not require holding an additional device.

3 DIGITOUCH SYSTEM

In this section, we discuss the design of interactions enabled by DigiTouch, followed by the hardware and software implementation of the system.

3.1 DigiTouch Interactions

DigiTouch interactions are based on thumb-to-finger touches. It is a continuous-input mechanism that allows the thumb to manipulate virtual widgets placed along the fingers. This is enabled by the unique flexibility of the thumb, whose adduction and flexion ability enables it to comfortably touch the other four fingers. It provides both tactile and proprioceptive cues to the user, enabling eyes-free accurate touch locations. Also, as each finger is separated into three phalanges, the joints between these segments serve as natural reference points for the user. *DigitSpace* [11] explored the comfort and accuracy of thumb-to-finger touches. They found that users can comfortably and accurately discriminate between up to five buttons on most fingers, in an eyes-free manner. We leverage these findings to inform our interaction design.

We explore interactions that rely on the following touch gestures: *tap*, *force tap*, *slide*, and *chording* (touching two fingers of opposite hands simultaneously). These simple gestures act as building blocks and enable a wide range of functionality, including typical buttons, pressure-sensitive buttons, sliders, chording input, and even a full keyboard for text input. DigiTouch's continuous sensing capability ensures that the input is reconfigurable and various input controls (such as buttons, sliders, etc.) can be mapped to finger positions according to the

---

[3] http://twiddler.tekgear.com/

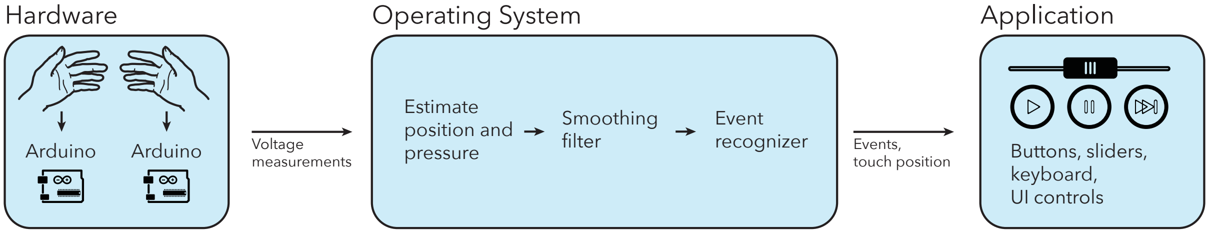


Fig. 3. Overview of DigiTouch tracking system. The hardware layer measures voltages from the gloves and transmits them to a computer for processing. The operating system layer on the computer estimates the touch position and pressure, smooths the signals, and recognizes the event. Finally, applications can implement user interfaces on top of this layer.

needs of an application. For example, a music player may use only a few buttons per finger on a single hand (Figure 2, (center)), whereas a 3D modeling application may use fine-grained controls with up to five buttons and/or sliders per finger.

3.2 DigiTouch Glove Design

DigiTouch is implemented as a glove-based system since gloves provide the most reliable way to detect contact between the thumb and finger. Other tracking techniques may be able to provide more accurate hand pose estimation or positional tracking, but are less reliable at detecting actual contact events. Our system consists of two gloves with a touch strip along the length of each of the four fingers and a conductive patch on the pad of the thumb. Each glove is powered by an Arduino Trinket Pro which streams data to a computer over a wired serial connection. Though not used in the evaluation, we also implemented a wireless version of the gloves that streams data using the GZLL wireless protocol. Software on the computer receives, processes, and filters the data to determine touch position and pressure on each hand. Based on this information, it triggers events that contain information about the touch position and pressure and describe the hand's state, such as *OnThumbDown* and *OnThumbMove*. It then sets up a web socket for any web-based client to receive the touch events. We implemented clients on a HoloLens as well as a standard web browser. Figure 3 shows a high-level overview of the DigiTouch system.

While constructing the DigiTouch glove, our main design considerations were: the glove fabric should be thin and elastic, ensuring good contact with the skin and preserving tactile feedback, and the glove should be comfortable, so that it can be worn for extended periods of time. DigiTouch (Figure 4) consists of three main components: (1) the thin elastic nylon glove, (2) partially conductive fabric strips that act as linear touch sensors on four fingers, and (3) conductive fabric to make a thumb patch. We chose to use conductive fabric over other conductive materials because it is flexible and maintains its conductivity over time. The partially conductive fabric strips are made from a polyester/cotton blend with small stainless steel fibers to make it partially conductive[4]. The thumb patch is made from a cotton woven with stainless steel thread[5]. While we attach commercially available fabrics to an existing glove, we note that recent advances in digital textiles [25] will enable tighter coupling between the glove and the electronics. Future designs could have the conductive strips woven into the glove itself.

With this configuration, we ensure that the user can touch with any portion of the thumb, but the contact point on the finger strip determines the touch position. This is important because depending on anatomical differences, which finger is being touched, and the orientation of the thumb, a different part of the thumb will

---

[4] Staticot, from Less EMF
[5] HertzCloth, from Less EMF

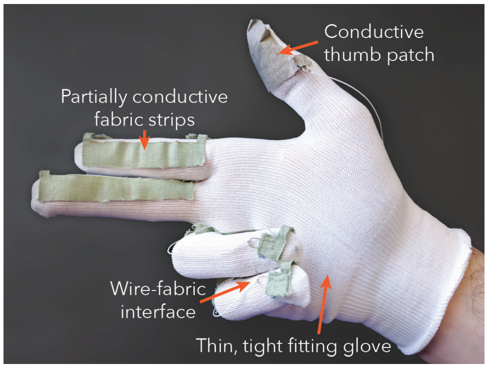


Fig. 4. DigiTouch consists of a conductive thumb patch that makes contact with one of the partially conductive fabric strips along the fingers.

make contact with the finger. Thin wires are attached to each end of the finger strip and one end of the thumb patch using conductive silver epoxy (Figure 4). This provides a reliable wire-fabric interface that does not change with movement.

3.3 DigiTouch Sensing

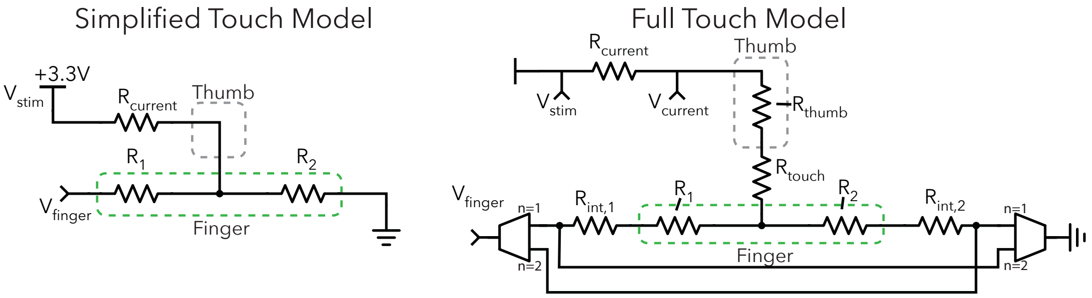


Fig. 5. (left) Simplified circuit model of the thumb and finger circuit. Current flows through $R_{current}$, through the thumb, and makes contact with the finger between R_1 and R_2. (right) Full circuit model showing all voltage measurements and multiplexed input. Each end of the finger is alternately ($n = 1, 2$) grounded while the voltage at the other end is measured. Since touching with increased pressure reduces the resistance at the fabric-fabric interface, estimating R_{touch} gives an estimate of touch pressure.

The fabric strip on the fingers has a resistance that is approximately uniform across the length of the strip. Each strip has a total resistance of approximately 500 Ω. We model each finger as a potentiometer, and the thumb acts as a wiper that slides along the resistive fabric. Figure 5 (left) shows a simplified schematic of the glove with one finger and one thumb. When the thumb makes contact with a portion of the finger: R_1 represents the resistance of the partially conductive fabric between the knuckle and thumb, and R_2 represents the resistance of the partially conductive fabric between the thumb and fingertip. A voltage, V_{stim} (3.3 V, nominal), is applied to the thumb patch and $R_{current}$ is a constant physical resistor that limits current flow and allows us to treat the circuit as a voltage divider. By measuring the voltage at the base of the finger (V_{finger}) and connecting the fingertip to ground, we can estimate R_2 according to Equation 1.

$$V_{finger} = V_{stim}\frac{R_2}{R_{current} + R_2} \tag{1}$$

Assuming that the overall resistance of the fabric ($R_T = R_1 + R_2$) is constant, we can estimate the touch location, x according Equation 2.

$$x = \frac{R_T - R_2}{R_T} \tag{2}$$

Unfortunately, this simple design presents several challenges. First, the overall resistance of the cloth (R_T) changes significantly with finger bending/stretching, and over time due to environmental factors. Second, since the conductive thumb pad contacts the finger over an area, the overall resistance between the two ends of the finger strips is reduced. Third, this model assumes the user makes perfect contact between the thumb and the finger; however, inconsistent touch pressure can result in additional resistance.

We took several steps to address these challenges, thus estimating touch pressure and accounting for dynamic changes in fabric resistance. Figure 5 (right) shows the full circuit design with one finger and one thumb. In this model, we account for additional resistances at each of the wire-fabric interfaces ($R_{thumb}, R_{int,1}, R_{int,2}$) and the fabric-fabric interface resistance at the touch location (R_{touch}). We also use an eight-channel digital multiplexer and make two sets of measurements to account for the variable resistance in R_T, due to bending/stretching of fingers. In Figure 5 (right), only two of the eight channels are depicted. The remaining channels are connected to the other fingers in a similar manner. When the user's thumb is not touching a finger, the microcontroller toggles between the eight multiplexer channels through three digital logic lines. By toggling these, the microcontroller rapidly switches the ground and analog-to-digital converter (ADC) connections between the tops and bottoms of the four fingers, such that when one side of a finger strip is grounded, the other is connected to the ADC. As soon as a touch is detected, it switches to a focused-mode where it rapidly toggles only between the top and bottom of the finger being touched to increase the data rate. The non-grounded end is measured with an ADC (V_{finger}). We also measure current flowing through the thumb by measuring the voltage drop across the current limiting resistor.

To summarize, for each finger, we measure the following: $V_{stim,1}$, $V_{current,1}$, and $V_{finger,1}$, when the inside of the finger is grounded and $V_{stim,2}$, $V_{current,2}$, $V_{finger,2}$ when the outside of the finger is grounded. R_{thumb}, $R_{int,1}$, and $R_{int,2}$ are constants that can be set with a one-time calibration.

We first estimate the current flowing through the thumb for each state ($n = 1$, $n = 2$) according to Equation 3. We then compute the resistance of each section of the finger strip (R_1, R_2) according to Equation 4. We compute touch position as before (Equation 2).

$$I_n = \frac{V_{stim,n} - V_{current,n}}{R_{current}} \tag{3}$$

$$R_n = \frac{V_{finger,n}}{I_n} - R_{int,n} \tag{4}$$

Figure 6 shows an example of how our model accounts for the varying resistance of the fabric strips during a typical swipe gesture. In a typical voltage divider model, when the electrical properties of the fabric change over time due to mechanical changes in the fabric or sweating of the user, it causes the total resistance to drift. With our time-multiplexed approach, we no longer depend on knowing the total resistance of the fabric strip and we can more consistently estimate the touch position.

Finally, we estimate the pressure between the thumb and finger by computing R_{touch} according to Equation 5. We model pressure as the resistance between the fabric-fabric interface. As the user presses harder, both the contact area and the conductivity of the contact area increase. We compute the pressure estimate twice, using each set of measurements ($n = 1, 2$) independently and average them together. The pressure output is computed according to Equation 6, where k is a constant set to the minimum possible touch resistance.

$$R_{touch,n} = \frac{V_{stim,n}}{I_n} - R_{current} - R_n - R_{extra,n} - R_{thumb} \tag{5}$$

$$Pressure = \frac{k}{\max(k, \frac{R_{touch,1}+R_{touch,2}}{2})} \tag{6}$$

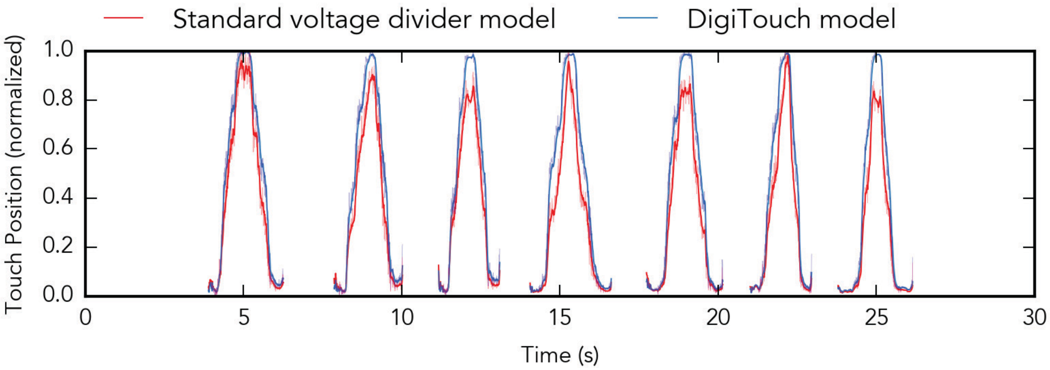


Fig. 6. A series of swipes starting at the base of the finger (position = 0), moving to the tip of the finger (position = 1), and returning to the base (position = 0). Using the traditional voltage divider model (red) with ground at the fingertip, the total resistance drifts over time. In DigiTouch we measure the finger resistance from both the fingertip and the finger base, accounting for any resistance drifts over time.

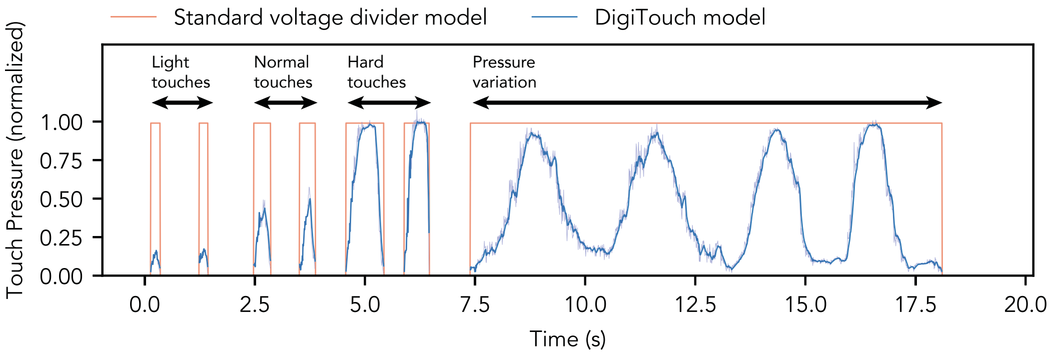


Fig. 7. DigiTouch pressure signal when a user performs two light touches, two medium touches, and two hard touches. Finally the user maintained contact and varied the contact pressure. The DigiTouch model enables force-sensitive interaction, while a standard voltage divider model would only detect a binary touch event.

Figure 7 shows an example of the pressure signal measured by DigiTouch. In this example, the user performs a series of light, normal, and hard touches on the index finger, followed by ten seconds of continuous contact while varying the pressure. This illustrates the ability to enable force-sensitive interactions, such as those enabled by the 3D touch features on iOS devices.

3.4 DigiTouch Signal Processing

Raw data is processed by software on a computer that streams data from the glove over two serial port connections. Though this implementation uses a wired setup, it could be easily made wireless using Bluetooth or a similar protocol. Each message frame contains information about which finger was touched and the six voltage measurements describing the touch state. During a touch, data is sampled and processed at 160 Hz. At all other times, the data stream is monitored at 70 Hz. For each set of measurements, the software computes the contact position and touch pressure, and dispatches relevant events that took place since the last measurements. The event model includes *OnThumbUp*/*Down* events, an *OnThumbMove* event, and *OnBeginForce*/*OnEndForce* events. Force-based events are triggered when the user's touch pressure exceeds a threshold.

Because of inconsistent touch contact and movement of the fingers, the raw signal can be noisy. We therefore pass all six voltages signals through an exponential smoothing filter ($\alpha = 0.2$) to smooth them before any computation is performed. At the start of a touch event, these filters snap to the first detected set of measurements, to minimize lag. These smoothed values are then used to compute the touch position and pressure, as previously described. Since the signal changes rapidly when the user first touches their thumb to a finger, we implemented a second stage exponential smoothing filter for the position and pressure signals. This filter uses a dynamic smoothing factor that starts high ($\alpha = 1$, no smoothing) when the user first touches and decreases throughout the touch event (α drops 0.2 per frame, down to a limit of $\alpha = 0.2$). These parameters are reset on every touch. This helps minimize the effects of drift when the user first makes contact and makes the system feel more responsive.

We use a finite state machine to recognize events. The state machine keeps track of the current conceptual state of the user's interaction with the glove ("not touching", "touching but holding still", or "moving", in addition to how hard the user is pressing). Event transitions are determined by the changes in the position (for *OnThumbMove*) and pressure (all other events) signals. An event is fired every time the machine transitions, along with important details associated with the event (such as position for an *OnThumbUp* event, or starting and ending positions for an *OnThumbMove* event). The sequence of events that are triggered for several user actions are:

- Tap: *OnThumbDown* → *OnThumbUp*;
- Force Tap: *OnThumbDown* → *OnBeginForce* → *OnEndForce* → *OnThumbUp*;
- Slide: *OnThumbDown* → *OnThumbMove* → *OnThumbUp*.

Finally, the detected position, pressure, and events are written to a WebSocket, ready for consumption by a web-based user interface widget.

4 TEXT ENTRY KEYBOARD DESIGN

We chose text entry using a full QWERTY keyboard to study the performance of DigiTouch in a real-world task. Text entry is a complex task requiring more than 26 buttons, which is hard to achieve using most input modalities. Most devices designed for wearable input rely on chording [16, 17, 27] due to a limited number of buttons. This leads to an increased cognitive load as users learn to use the system. On DigiTouch, we created 28 keys (26 letters + *space* + *backspace*) accessible by at most one tap on the finger using the thumb. The keyboard layout is illustrated in Figure 2 (right).

We chose the standard QWERTY keyboard as it is most familiar to people. In a QWERTY layout, there are at most ten letters in each row. By splitting the QWERTY keyboard in two halves, we ensure there are never more than five buttons assigned to each finger. In this layout, each thumb acts as a *stylus* and effectively operates on one half of the keyboard. Thus, it closely resembles two-thumb typing on a smartphone or tablet. In fact, many users are already familiar with a split-keyboard layout, since it is a feature on both Android and iOS for tablets. Because of this familiarity, we expect a smaller learning curve compared to chording-based text entry systems.

Huang et al. [11] estimated the maximum number of buttons that can be placed on each finger before the target selection performance becomes unusable. They reported that users could trigger five buttons per index and middle finger, four buttons per ring finger, and two buttons per pinky finger, with high accuracy. Our keyboard layout falls within their suggested guidelines.

The space and backspace keys are the two most commonly used keys on a keyboard [36]. To avoid frustration for users, they must not be confused with other keys. We decided to dedicate the left and right pinky fingers to the Space and Backspace keys, respectively (Figure 2, right). In the future, one can imagine adding control or punctuation keys to the outsides of pinky fingers.

Though tactile and proprioceptive feedback helps in target selection, users might not always press the correct target initially, particularly on fingers with four or five buttons. Because DigiTouch can track the continuous position of the thumb sliding along the finger, we mimic the interaction on modern smartphone keyboards and allow the user to slide their thumb along the fingers to select the correct key. On a thumb-down event, a letter gets highlighted. The user is free to slide the thumb on the finger to switch to adjacent letters. When the user's thumb is over the letter that they wish to press, a thumb-up event triggers letter selection.

5 TEXT ENTRY EVALUATION

5.1 Study Design

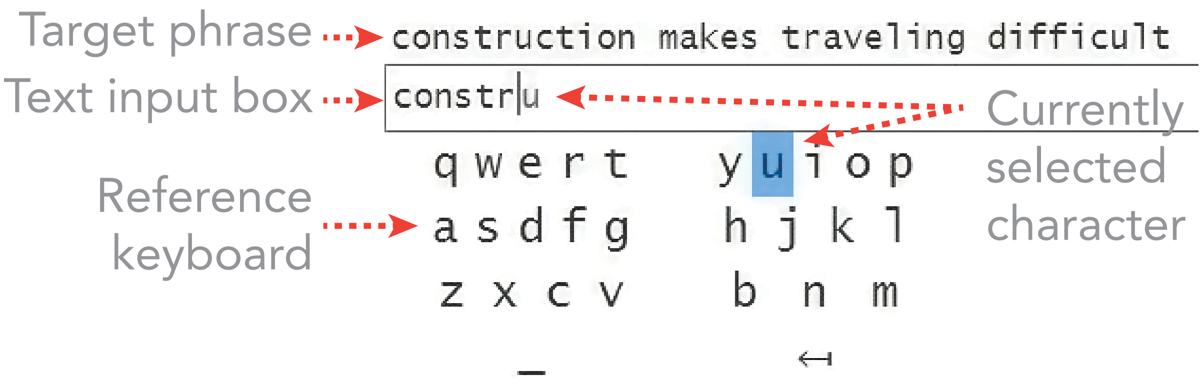
The goals of our controlled evaluation was to observe: (1) whether the participants were able to type using the thumb-to-finger interactions enabled by DigiTouch; (2) how their performance varied over time and with practice; and (3) whether personalized keyboard model impacts performance.

5.1.1 Participants. Ten participants (7 male, 3 female), with a mean age of 23.1 years (18 - 27 years) participated in the study. Eight were right-handed (two left-handed), and all self-rated as expert touch screen smartphone users. Each participant was compensated $5 per twenty-minute session.

5.1.2 Apparatus. Participants interacted with our custom text entry software (Figure 8) running in a web browser (Google Chrome) on a Windows 7 desktop computer. Participants sat on an adjustable reclining chair and were asked to keep their hands in any comfortable position. In order to evaluate eyes-free input, they were simply asked to position their hands such that they are unable to see their hands while typing (by keeping their hands in their lap or by putting their hands under the table). The software logged all of the users' touch events.

5.1.3 Procedure. The procedure was designed to fit in ten 20-minutes sessions spread over a period of 15 days, with consecutive sessions separated by at least six hours and no more than two days. However, due to scheduling conflicts, some participants exceeded this two-day limit. For each participant, the first session began with an introduction to DigiTouch, the text entry task and the experiment software. At the beginning of each session participants were asked to swipe along each strip to ensure proper connection and fit. Occasionally, a wire would come loose during a session due to strain. In such events the session was paused while the connection was restored, and the current phrase was thrown out.

For each 20-minute session, participants were asked to input as many phrases as possible, similar to [5]. The phrases were randomly chosen from a published phrase set for text entry by MacKenzie and Soukoreff [20], with average phrase length of 28.61 characters. The experiment was conducted only with the lower case letters (no upper case letters, punctuation, or numbers). All the participants received the phrases in a randomized order. We asked the participants to type "as quickly and accurately as possible", and to fix errors unless those errors were made "far behind" their current point of entry. Participants were encouraged to take a short break between



construction makes traveling difficult

constru

Fig. 8. Custom web application for the text entry user study. Participants were presented with a target phrase and instructed to type it as quickly and accurately as possible. Participants could see their input beneath the target phrase. A reference keyboard showed the key placement on the fingers and highlighted the currently selected key.

phrases, anytime they wished. The user interface (Figure 8) showed phrase to be transcribed (target phrase), a text box to receive participants' input, a timer, and a 'Pause' button at the bottom. After completion of a phrase, participants used a chording gesture by simultaneously touching both thumbs to their respective index finger to move on to the next phrase.

5.2 Text Entry Results

All ten participants completed ten sessions. In total, participants entered 3686 phrases. The main measures for evaluating the performance of DigiTouch were typing speed, corrected error rate, and uncorrected error rate. We also conducted an analysis of the input stream to arrive at character level metrics that characterize performance of different regions of the glove. Unless otherwise noted, all reported means and standard deviations are computed across participants and phrases.

5.2.1 Speed. Text entry speed was measured in words per minute (wpm), calculated as (characters per second)× $\frac{60}{5}$, using the assumption that a word consists of 5 characters [19]. The average text entry speed over all sessions and across all participants was 13.0 wpm ($\sigma = 4.12$). The mean speed for the first session was 7.0 wpm ($\sigma = 2.2$), while the mean speed for session 10 was 16.0 wpm ($\sigma = 4.1$), clearly showing that performance increased with practice (Figure 9). The learnability curves obtained (Figure 9) are similar to the characteristic learnability curve for text entry system [17].

Speed at a character level was analyzed using *dwell time*, measured as the time difference between the thumb down and the following thumb up event. Dwell time for the first session, averaged over all the participants was 443 ms ($\sigma = 124$ ms), while for the final session, it was 301 ms ($\sigma = 74$ ms). This shows that time taken to input a character reduces with practice over time. Figure 10 shows the average dwell time for each correctly entered character during the final session. Lower dwell time for the characters situated on the extremes suggest users are more confident in pressing these characters.

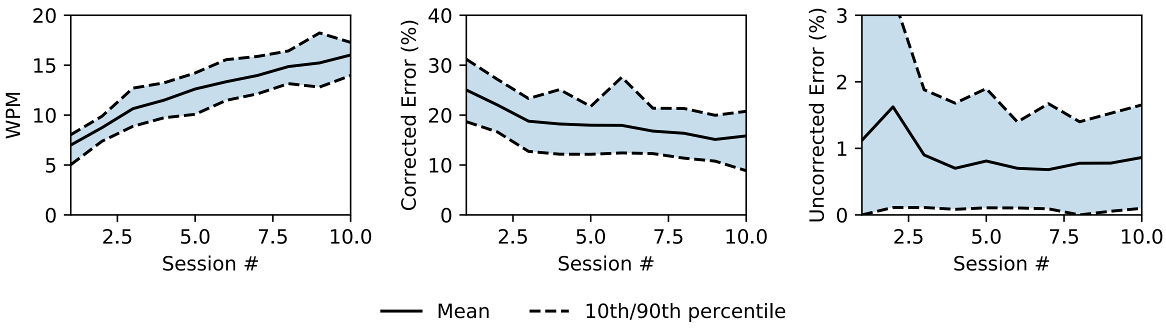


Fig. 9. Summary of text entry study results. (left) Typing speed measured in words per minute (wpm) increased with practice. (center) Corrected error rate, a measure of errors that users eventually corrected, decreased with practice. (right) There were no clear trends in the uncorrected error rate, a measure of errors remaining in the final transcription.

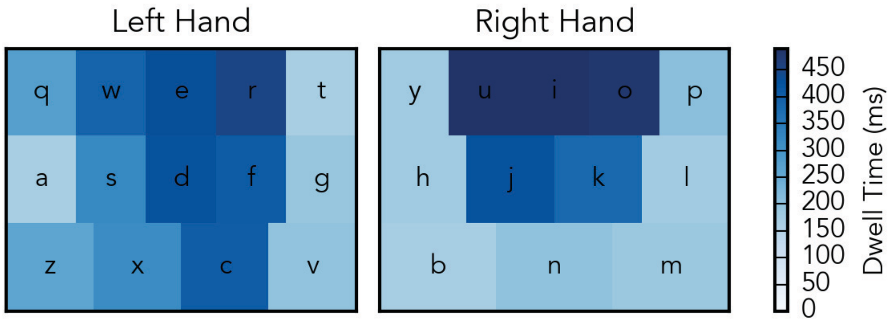


Fig. 10. Heatmap showing mean dwell time for each correctly pressed key. The time indicates how long the user held the thumb down to eventually type each character.

5.2.2 Accuracy. Two metrics were used to measure text entry accuracy: (a) Corrected error rate [30] – a measure of the errors that the user corrected in the final transcription, and (b) Uncorrected error rate [30] – a measure of the errors remaining in the transcribed text that the user did not correct. A user's typing speed represents a trade-off between corrected and uncorrected errors. More corrections result in a slower typing speed, as each correction adds multiple keystrokes, i.e., backspace character, re-enter character. These metrics were computed using software developed by Wobbrock et al. [36].

The mean corrected error rate for session 1 was 25.0% (σ = 11%), and for session 10 was 15.8% (σ = 10%). For the last session, this means that 15.8% of all characters typed were ultimately incorrect characters that the user fixed. The mean uncorrected error rate across all sessions was 0.85% (σ = 2.3%), and there was no clear trend across sessions (Figure 9, right). This suggests that participants' tolerance for error showed little variation with practice. Since the corrected error rate dropped and was accompanied by no distinct increase in uncorrected error rate, this suggests that the number of errors users had to correct decreased with practice (Figure 9, center).

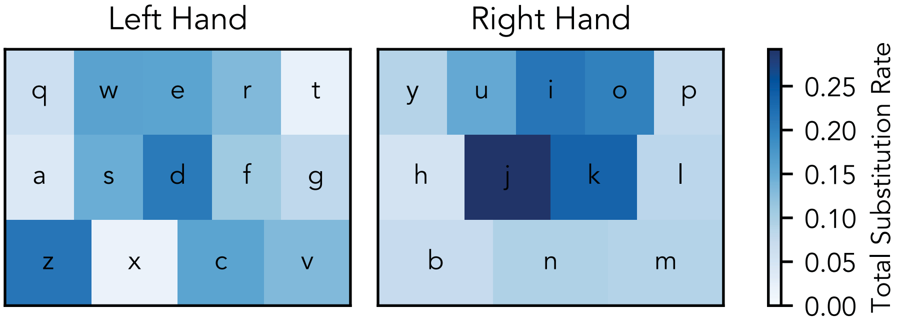


Fig. 11. Heatmap showing total substitution rate for each key. Higher errors indicate less accuracy in attempting to type a particular character.

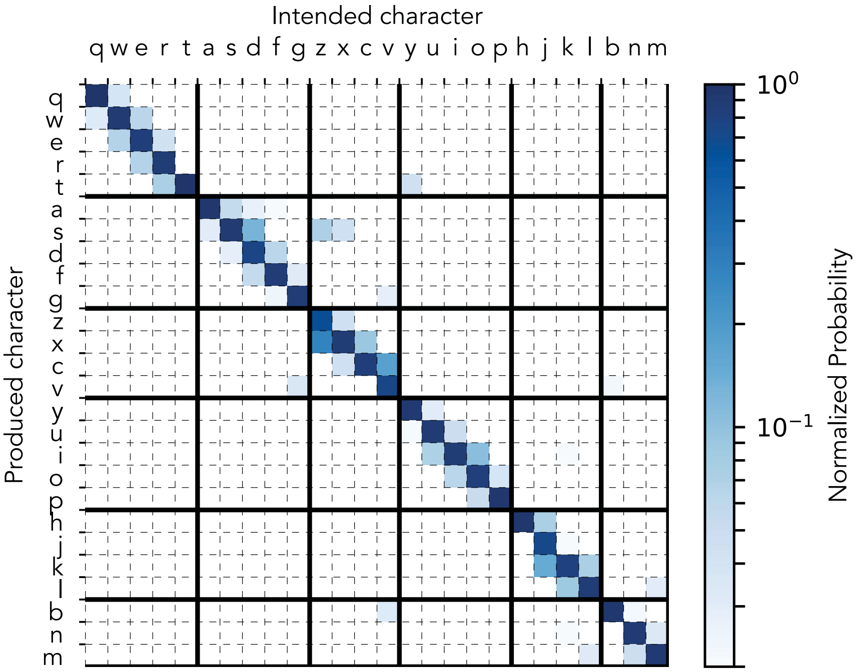


Fig. 12. Confusion matrix showing the probability of produced vs intended characters according to an analysis of the input stream. The plot is ordered by the placement of keys along the finger so that the most often confused letters appear closer to the diagonal. The scale is logarithmic to better highlight the confusions.

5.2.3 Input Stream Analysis. These error rates consider participant's accuracy on a phrase level, but more nuanced results can be obtained from an analysis of the input stream. We use the approach developed by

Wobbrock et al. [36] to compute all possible alignments between the input stream and the intended output. The *character-wise total substitution rate* (Figure 11) answers the question, "when trying for *i*, what is the probability that participants did not get *i*?". This allows us to examine which characters were most prone to confusion. In general, participants were more accurate on characters situated on the extremes of the fingers, and characters situated on fingers with fewer buttons. The highest substitution rate was seen for the characters 'j' and 'z'. This may be due to the fact that they occurred very few times in the target phrase ('j' 38 times, 'z' 8 times, compared to 'e' 1480 times), thus making these errors less significant.

A confusion matrix represents the frequency of character-level errors [18]. Figure 12 shows the total number of times when an intended character was transcribed with a (correct/incorrect) produced character. The most prevalent mistakes were: transcribing 'i' instead of 'o', 'k' instead of 'j', etc. Since most of the errors in the confusion matrix are adjacent letter confusion, an auto-correct can dramatically boost the text entry performance.

6 PRESSURE EVALUATION

While the text entry study evaluates the touch sensing capabilities of DigiTouch, it does not exercise the pressure sensing functionality. To evaluate the pressure sensing capabilities of DigiTouch, we conducted a short user study with 10 participants (6 male, 4 female) with a mean age of 21.9 years (18 - 25). The experiment was split into two phases and took approximately 10 minutes. In the first phase, participants were asked to touch the thumb to the index finger with either *light* or *hard* pressure. In the second phase, a third level was added (*light*, *medium*, and *hard*). In both phases, the order was randomly presented and randomly switched between the left and right hand to minimize fatigue. The user was presented with the type of touch to perform and after a 1 second delay to prevent users from rushing, a tone was played to indicate that the user could perform the touch. When an up event was detected, the pressure recorded was the maximum detected pressure during the touch event.

The pressure boundaries between levels were set based on the results of a pilot study that was conductive with a separate set of five participants. Before each phase, participants first learned the pressure boundaries during a practice period in which they received visual feedback on the pressure. Once participants performed the tap with the correct pressure, they were allowed to move on. During the actual experiment, participants performed the same task without receiving any kind of feedback. Participants performed 5 taps per level per hand for practice and 10 taps per level per hand during the evaluation.

In total, 1000 touches were collected from this study. DigiTouch identified the correct pressure on these touches with an accuracy of 93.3% in the two-level case (N=400) and 64.0% in the three-level case. The distribution of touches for each type of touch are presented in Figure 13. Hard touches were more tightly distributed while there was a broader distribution of light touches. Though we assume that the user always pressed with the intended touch pressure, we note that this may not always be the case, particularly for the three-level case. Several participants reported a conceptual difficulty in distinguishing between three different pressure levels. Others reported that they could perform the three-level task, but they had to concentrate harder. Users universally reported that the two-pressure task was easy. While applications may benefit from discrete pressure-enabled input, the relatively poor observed performance with three pressure levels suggests that reliably distinguishing more than two levels is difficult.

7 DISCUSSION

7.1 Glove Design

Head-mounted displays have the potential to enable truly ubiquitous computing. Though a universal input solution is difficult because different situations demand different capabilities, a user should be able to use the right technique for the situation. Prior attempts to add input to gloves have produced bulky systems with low adoption. Our sensing technique enables continuous input with relatively little instrumentation, compared to

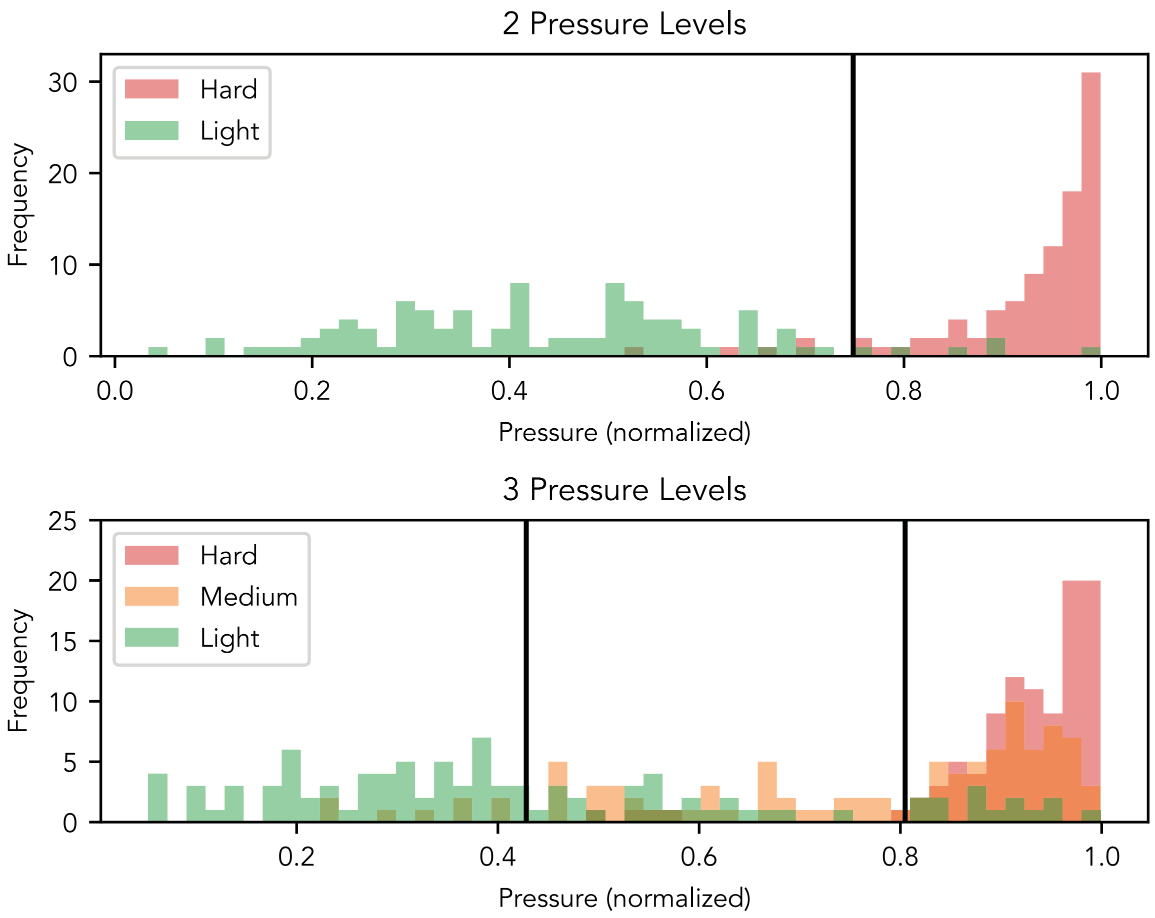


Fig. 13. Distribution of measured touch pressures when asked to touch either light or hard (top) or light, medium, or hard (bottom). Note that the distributions are drawn partly transparent in order to view the overlapping distributions underneath. The black lines indicate the predefined boundaries between pressure levels.

other glove-based devices. To achieve this style of continuous tracking with traditional techniques would require significantly more instrumentation, using multiple fabric layers or many wired contact points. Moreover, AR and VR present use cases for gloves that may overcome prior hurdles to adoption. For example, certain capabilities, such as high-fidelity haptic feedback, are likely impossible without the use of gloves. Also, in cases where precise hand tracking is required, gloves can simplify the task of pose estimation [35].

DigiTouch is particularly compelling when combined with traditional optical hand tracking techniques, which excel at pose estimation, though are not precise enough to detect thumb touch events or touch locations. With the hand pose and segmentation from a hand tracker, the virtual widget layout can be superimposed directly on the user's fingers. For example, in a text entry system, a user might see the individual characters superimposed on their fingers at the correct location. This would significantly improve the learnability of such a system.

With any wearable system, there is a concern for false activations. This is particularly important with any kind of hand-based sensing, since we use our hands for so many things. These issues will arise primarily in an AR scenario. We envision that such a system can have an activation gesture that enables continuous position and pressure sensing on all fingers. With DigiTouch, a number of different activation gestures are possible. For example, users could tap on each finger sequentially or swipe along a particular finger to enable input.

One challenge we observed with thumb-to-finger input is determining the precise location of a touch point. When users touch their thumb to a finger, they make contact at an area and it can be unclear where the intended touch point is, as it depends on the target finger and the orientation of the thumb. In an initial prototype of the DigiTouch glove, we used a small thumb patch to minimize the ambiguity in the touch location. However, we found that this caused significant frustration for users since they would often miss the finger strips completely. As a result, we settled on a large patch that covers most of the thumb in order to maximize the reliability of touch detection at the expense of precision in estimating touch position. As one might expect, this also causes some issues with drifting touch locations as the user rolls and lifts the thumb from the finger at the start and end of touch events. In DigiTouch, we counter this drift by modifying filter parameters to minimize drift at the start and end of touch. In *Understanding Touch*, Holz and Baudisch explore a closely related problem with touchscreens on mobile devices [9]. With additional study, it may be possible to construct a model for intended touch location as a function of the thumb position and contact area. This may enable a dynamic correction that could significantly improve touch precision.

7.2 Using DigiTouch for Text Entry

With any new input system, there are many aspects to evaluate. For DigiTouch, we chose to evaluate one of the most challenging use cases that covers most of the interaction space. In our proposed keyboard design, users tapped and swiped to input a character, and used chording to advance to the next phrase. We use the ability of users to input text in this manner to claim that users would also be effective at controlling other applications with custom layouts.

The results from the text entry evaluation demonstrate that users were able to effectively type using DigiTouch. User performance in our study exceeded that of many other similar wearable text entry systems. Rosenberg et al. [27] showed a mean typing speed of 8.9 wpm after 80 minutes and 16.8 wpm after nearly 10 hours of practice. DigiTouch achieves a similar speed after only 3.3 hours of practice. The input glove designed by Hsieh et al. achieved a mean typing speed of 5.4 wpm after twenty minutes of practice [10]. In the same amount of time, DigiTouch users achieved a similar typing speed of 6.5 wpm. Lyons et al. evaluated the learnability of Twiddler [6], a commercial product often used for text entry on wearable systems. They found that the mean text entry speed was 19.5 wpm after 200 minutes, and increased to 47 wpm after 25 hours of practice [17]. Though our study design was limited to 200 minutes of practice, DigiTouch performed comparably at 16.5 wpm (vs 19.5 wpm). By looking at the typing speed progression over sessions (Figure 9, left), it is hard to judge improvement beyond the initial 10 sessions. However, the continuous nature of DigiTouch enables additional controls like application specific layouts and sliders, something that cannot be achieved with a Twiddler.

Most users during the study chose to hold their hands apart, with their arms resting on the chair or at their side. This suggests that users were comfortable with the mental model of the split keyboard. It also highlights an advantage of DigiTouch over other hand-based interaction techniques that requires the hands to be held up. DigiTouch can easily be used with the arms at rest and may even be appropriate for use while walking.

During the study, one participant noted that after thinking of the system as a smartphone held in landscape mode, their performance improved significantly. This highlights the importance of using a familiar layout for the

---

[6] http://twiddler.tekgear.com/

users. However, expert users may wish to customize the layout to put the most commonly used keys in the most accurate and comfortable regions.

In a text entry system, a user's typing speed is related to the number of errors they correct, as correction takes time away from inputting the desired text. In our study, the corrected error rate was around 15% in the last session. This is somewhat high compared to other text entry systems. These errors are a combination of a user's inaccuracy in touching the correct part of their finger and DigiTouch's inability to accurately sense the intended touch location. For user's touch accuracy, prior work suggests that users are able to accurately perform thumb-to-finger touches [11], but they did not examine whether this accuracy holds up when typing rapidly. Similarly, for DigiTouch's sensing accuracy, a high-accuracy motion capture system, such as OptiTrack[7], can help in decomposing the sources of error.

Though our evaluation of DigiTouch used a fixed keyboard layout, the continuous sensing could enable an adaptive keyboard model. Users with different sized hands or a different range of motion of the thumb may prefer a condensed or expanded layout, for example. With discrete buttons, it is impossible to relocate them or create dynamic touch regions. During the study, several users expressed frustration at the difficulty of pressing a particular key (usually on a finger with four or five buttons). With an adaptive model, these keys could be virtually expanded to make them easier to hit.

Most modern keyboards offer intelligent auto-completion and auto-correction features. Though we evaluated DigiTouch using text entry with auto-complete functionality disabled to enable comparisons with related work, we anticipate that this can significantly improve typing speed and accuracy. To explore this, we built a novel auto-suggestion system that utilizes the pressure sensing capability of DigiTouch. We assign one potential word completion to each finger. A user can trigger a word completion with a force press on the corresponding finger. The high accuracy on the two-level pressure study supports the feasibility of this approach. Though we do not formally evaluate the performance of these features, one of the authors was able to achieve an average speed of 30 wpm using it.

Finally, while we evaluated the QWERTY layout because it was easy to learn and provided a rigorous test of DigiTouch, we note that power users can achieve performance gains by using more advanced keyboard layouts. More commonly used letters can be moved to the outsides of the fingers and can be spaced further apart to significantly boost performance. Less frequently used keys could be relegated to two-touch keys that require chording.

7.3 Limitations and Future Work

For wearable systems, the design and construction has a significant impact on user performance. To maximize comfort and minimize bulk, DigiTouch uses a thin one-size-fits-all elastic glove. However, the touch strips we add to the fingers are not elastic. Though all participants found the glove to be comfortable, some participants mentioned that they would have preferred a glove tailored to their own hand size. Anecdotally, we did not see any deterioration in performance for these participants, but in the future, it may be beneficial to design gloves in several sizes to accommodate users with different hand sizes.

Though DigiTouch is designed for wearable AR and VR scenarios, our evaluation is conducted at a desktop computer. Initial pilot testing with a VR system introduced additional factors unrelated to the performance of DigiTouch, such as general unfamiliarity with VR systems and discomfort due to extended use of a head-mounted display. DigiTouch is capable of handling varying resistance of the fabric strips, but it assumes a uniform resistance along the length of the strip at any point of time. This can become problematic near each end, where the wire connection is made on the sides of the finger (Figure 4). Because of this connection, the end points are more conductive, which results in a nonlinear region at the extremes. Future versions should be able to account

---

[7] https://www.optitrack.com

for this with a one-time calibration. DigiTouch enables continuous pressure input. This is slowly becoming popular on consumer smart devices, and perhaps will be even more valuable in the 3D environments of AR/VR systems. In the future, it will be interesting to evaluate the benefits and performance of pressure input, beyond discussing its use for auto-correction in text entry.

To explore the use of DigiTouch in an AR device, we created a wireless version of the gloves that streams data to a Unity application running on a HoloLens. We designed a custom Unity input module that allows DigiTouch to be used with all the standard Unity UI components, including buttons and sliders. Using this module, we constructed two example experience that can be controlled with DigiTouch: a music player and a text entry system. While these experiences highlight the potential of this style of input with AR systems, we leave a formal evaluation to future work.

DigiTouch's continuous position and pressure input may support richer interactions when combined with other sensing. For example, in a sculpting application, a user might configure a brush with color, brush size, and other settings, then force tap it onto a finger so it "sticks." Future taps on that finger would trigger that tool, enabling user-customized interfaces. Using gaze selection, a user might look at a virtual object and drag along the finger to move it closer or farther away; added hand orientation sensing might set the direction of movement. Adding other sensors, such as an inertial measurement unit or flex sensors, could enable even richer interaction that takes advantage of the movement and posture of the hand. Adding vibrotactile haptics could improve feedback and add affordances. For example, it may vibrate when a user slides across a button that can to be pressed.

8 CONCLUSION

DigiTouch is a reconfigurable glove-based input device for wearable computing, particularly head-mounted AR and VR systems. DigiTouch enables subtle thumb-to-finger interactions by sensing the continuous touch position and pressure of the thumb along the fingers. To achieve this, we present a novel technique using only partially conductive fabric and a two-wire interface on each finger, with a conductive fabric patch on the thumb. Continuous touch tracking enables a set of easily reconfigurable widgets, which can be customized based on user preferences and application needs. To evaluate the performance of DigiTouch in a real-world application, we conducted a longitudinal text entry study using split-QWERTY keyboard. Participants achieved a mean typing speed of 16.5 wpm with high accuracy, showing the feasibility of using DigiTouch for text entry. The subtle, yet always-available input offered by DigiTouch has the potential to enable broader use of AR and VR systems.

ACKNOWLEDGMENTS

The authors would like to thank Spencer Pearson, Laura Trutoiu, and James Fogarty for their helpful ideas and discussions and Lilian de Greef and Nancy Thai for their assistance with illustrations and figures. The authors also thank the anonymous referees for their useful feedback and all the participants for their time, patience, and enthusiasm.

REFERENCES

[1] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. `DOI`: http://dx.doi.org/10.1145/2807442.2807450

[2] 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 (UIST '13)*. ACM, New York, NY, USA, 255–260. `DOI`: http://dx.doi.org/10.1145/2501988.2502016

[3] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 237–244. `DOI`: http://dx.doi.org/10.1145/2501988.2502035

[4] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1504–1514. DOI: http://dx.doi.org/10.1145/2858036.2858125

[5] Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. 2005. An Empirical Study of Typing Rates on mini-QWERTY Keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1288–1291. DOI: http://dx.doi.org/10.1145/1056808.1056898

[6] Sarthak Ghosh, Hyeong Cheol Kim, Yang Cao, Arne Wessels, Simon T. Perrault, and Shengdong Zhao. 2016. Ringteraction: Coordinated Thumb-index Interaction Using a Ring. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2640–2647. DOI: http://dx.doi.org/10.1145/2851581.2892371

[7] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 144–152. DOI: http://dx.doi.org/10.1145/2785830.2785867

[8] Florian Heller, Stefan Ivanov, Chat Wacharamanotham, and Jan Borchers. 2014. FabriTouch: Exploring Flexible Touch Input on Textiles. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC '14)*. ACM, New York, NY, USA, 59–62. DOI: http://dx.doi.org/10.1145/2634317.2634345

[9] Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2501–2510. DOI: http://dx.doi.org/10.1145/1978942.1979308

[10] 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 (CHI '16)*. ACM, New York, NY, USA, 4203–4215. DOI: http://dx.doi.org/10.1145/2858036.2858436

[11] 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*. ACM, 1526–1537.

[12] Hsin-Liu (Cindy) Kao, Artem Dementyev, Joseph A. Paradiso, and Chris Schmandt. 2015. NailO: Fingernails As an Input Surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3015–3018. DOI: http://dx.doi.org/10.1145/2702123.2702572

[13] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/2380116.2380139

[14] Marion Koelle, Matthias Kranz, and Andreas Möller. 2015. Don'T Look at Me That Way!: Understanding User Attitudes Towards Data Glasses Usage. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 362–372. DOI: http://dx.doi.org/10.1145/2785830.2785842

[15] 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 (VRST '05)*. ACM, New York, NY, USA, 86–95. DOI: http://dx.doi.org/10.1145/1101616.1101635

[16] Seongil Lee, Sang Hyuk Hong, and Jae Wook Jeon. 2003. Designing a Universal Keyboard Using Chording Gloves. In *Proceedings of the 2003 Conference on Universal Usability (CUU '03)*. ACM, New York, NY, USA, 142–147. DOI: http://dx.doi.org/10.1145/957205.957230

[17] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. 2004. Twiddler Typing: One-handed Chording Text Entry for Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 671–678. DOI: http://dx.doi.org/10.1145/985692.985777

[18] I. Scott MacKenzie and R. William Soukoreff. 2002. A Character-level Error Analysis Technique for Evaluating Text Entry Methods. In *Proceedings of the Second Nordic Conference on Human-computer Interaction (NordiCHI '02)*. ACM, New York, NY, USA, 243–246. DOI: http://dx.doi.org/10.1145/572020.572056

[19] I. Scott MacKenzie and R. William Soukoreff. 2002. Text Entry for Mobile Computing: Models and Methods,Theory and Practice. *Human-Computer Interaction* 17, 2-3 (2002), 147–198. DOI: http://dx.doi.org/10.1080/07370024.2002.9667313

[20] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. DOI: http://dx.doi.org/10.1145/765891.765971

[21] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1073–1082. DOI: http://dx.doi.org/10.1145/2556288.2556964

[22] Sam Miller, Andy Smith, Sina Bahram, and Robert St. Amant. 2012. A Glove for Tapping and Discrete 1D/2D Input. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces (IUI '12)*. ACM, New York, NY, USA, 101–104. DOI: http://dx.doi.org/10.1145/2166966.2166986

[23] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*.

ACM, New York, NY, USA, 201–204. DOI: http://dx.doi.org/10.1145/571985.572013

[24] 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 (ISWC '14 Adjunct)*. ACM, New York, NY, USA, 87–92. DOI: http://dx.doi.org/10.1145/2641248.2641266

[25] Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. 2016. Project Jacquard: Interactive Digital Textiles at Scale. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4216–4227. DOI: http://dx.doi.org/10.1145/2858036.2858176

[26] Manuel Pratorius, Ulrich Burgbacher, Dimitar Valkov, and Klaus Hinrichs. 2015. Sensing Thumb-to-Finger Taps for Symbolic Input in VR/AR Environments. *IEEE computer graphics and applications* 5 (2015), 42–54.

[27] R. Rosenberg and M. Slater. 1999. The Chording Glove: A Glove-based Text Input Device. *Trans. Sys. Man Cyber Part C* 29, 2 (May 1999), 186–191. DOI: http://dx.doi.org/10.1109/5326.760563

[28] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-available Input with Muscle-computer Interfaces. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/1622176.1622208

[29] Stefan Schneegass and Alexandra Voit. 2016. GestureSleeve: Using Touch Sensitive Fabrics for Gestural Input on the Forearm for Controlling Smartwatches. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 108–115. DOI: http://dx.doi.org/10.1145/2971763.2971797

[30] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 113–120. DOI: http://dx.doi.org/10.1145/642611.642632

[31] 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 (CHI '15)*. ACM, New York, NY, USA, 3643–3652. DOI: http://dx.doi.org/10.1145/2702123.2702136

[32] D. J. Sturman and D. Zeltzer. 1994. A survey of glove-based input. *IEEE Computer Graphics and Applications* 14, 1 (Jan 1994), 30–39. DOI: http://dx.doi.org/10.1109/38.250916

[33] Ying-Chao Tung, Chun-Yen Hsu, Han-Yu Wang, Silvia Chyou, Jhe-Wei Lin, Pei-Jung Wu, Andries Valstar, and Mike Y. Chen. 2015. User-Defined Game Input for Smart Glasses in Public Space. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3327–3336. DOI: http://dx.doi.org/10.1145/2702123.2702214

[34] 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 (MobileHCI '15)*. ACM, New York, NY, USA, 153–160. DOI: http://dx.doi.org/10.1145/2785830.2785886

[35] Robert Y. Wang and Jovan Popović. 2009. Real-time Hand-tracking with a Color Glove. In *ACM SIGGRAPH 2009 Papers (SIGGRAPH '09)*. ACM, New York, NY, USA, Article 63, 8 pages. DOI: http://dx.doi.org/10.1145/1576246.1531369

[36] Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the Input Stream for Character- Level Errors in Unconstrained Text Entry Evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (Dec. 2006), 458–489. DOI: http://dx.doi.org/10.1145/1188816.1188819

[37] Sang Ho Yoon, Ke Huo, Vinh P. Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-worn Textile Input Device with Multimodal Sensing in Mobile Interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York, NY, USA, 269–272. DOI: http://dx.doi.org/10.1145/2677199.2680560

[38] Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2014. Plex: Finger-worn Textile Sensor for Mobile Interaction During Activities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 191–194. DOI: http://dx.doi.org/10.1145/2638728.2638746

Received February 2017; revised May 2017; accepted July 2017