Thumb-to-Finger Gesture Recognition Using COTS Smartwatch Accelerometers

Riku Tsunoda
IPLAB
University of Tsukuba
Tsukuba, Japan
tsunoda@iplab.cs.tsukuba.ac.jp

Myungguen Choi
IPLAB
University of Tsukuba
Tsukuba, Japan
choi@iplab.cs.tsukuba.ac.jp

Buntarou Shizuki
IPLAB
University of Tsukuba
Tsukuba, Japan
shizuki@cs.tsukuba.ac.jp

Abstract

Thumb-to-finger gestures, such as 12 phalange taps, provide diverse input. This gesture enables eyes-free, rich, and one-handed smartwatch controls. Previous research on recognizing thumb-to-finger gestures has typically relied on additional sensors, making it difficult to achieve recognition using only a commercial off-the-shelf (COTS) smartwatch. In this paper, we developed a thumb-to-finger gesture recognition system that only uses the accelerometers built into a COTS smartwatch. Our recognition system achieved 80.1% accuracy for 17 gestures. Additionally, we developed optimal gesture sets for different numbers of gestures based on the recognition performance of the accelerometer. Consequently, we achieved 94.7% accuracy for 4 gestures in leave-one-participant-out cross-validation and 90.2% accuracy for 11 gestures in participant-specific leave-one-session-out cross-validation.

CCS Concepts

• **Human-centered computing → Gestural input**; **Interactive systems and tools**; **Mobile computing**.

Keywords

Wearable Device, Machine Learning, Finger Interaction, Vibration Sensing

ACM Reference Format:
Riku Tsunoda, Myungguen Choi, and Buntarou Shizuki. 2024. Thumb-to-Finger Gesture Recognition Using COTS Smartwatch Accelerometers. In *International Conference on Mobile and Ubiquitous Multimedia (MUM '24), December 01–04, 2024, Stockholm, Sweden.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3701571.3701600

1 Introduction

Smartwatches have become increasingly popular for monitoring health, receiving notifications, making phone calls, and controlling music players. However, due to their small size, the "fat finger problem" [47] is often encountered whereby unintended inputs occur because the user's finger obscures the target on the touch screen. Moreover, touch input can be impractical when one hand is occupied, such as when the user is carrying items [50]. It is also unsuitable for users with limited use of one hand. Although voice input does not have these limitations, it cannot be used in socially inappropriate places, such as quiet environments (e.g., libraries), and causes privacy concerns.

Gesture-based input using the smartwatch-wearing hand can overcome these challenges, as it does not require touching the screen and can be performed with one hand. Wrist gestures [12, 14, 26, 50] and hand gestures [6, 19, 41, 58, 63] are such methods. These features are available in AssistiveTouch[1] on the Apple Watch. However, these methods generally do not support eyes-free interaction, have a limited input vocabulary, or require additional sensors beyond the smartwatch.

Thumb-to-finger gestures represent a promising gesture-based input method using the smartwatch-wearing hand. These gestures involve interactions between the thumb and other fingers, providing a rich input vocabulary. For example, tapping different phalanges of each finger yields up to 12 gestures, enabling clear tactile feedback and a direct, fast, and discreet type of input [17, 48]. Thumb-to-finger gestures have several additional benefits. First, they do not require mode switching and can coexist with other touch gestures. Second, they are compatible with smartwatches or sports wristbands that lack a touchscreen, such as the Polar Pacer[2] and the Suunto 5 Peak[3]. Third, their subtle, hardly noticeable nature allows them to be performed naturally in public contexts in which conspicuous gestures may be perceived as socially awkward [1, 42]. Finally, the tactile feedback allows these gestures to be performed without looking at the smartwatch, enabling unobtrusive, "eyes-free" interactions [48]. Eyes-free gestures can also be effectively used to control devices in extended reality (XR) environments [15, 22, 43, 57].

However, recognizing thumb-to-finger gestures is challenging because they are small movements performed by body parts that are difficult to instrument [48]. Previous studies have explored various methods, such as attaching pressure sensors to fingers [56], using cameras on the wrist [40], shoulder [48], or chest [34], placing microphones and speakers on the back of the hand [25], attaching inertial measurement units (IMUs) to the fingers or the back of the hand [30, 52], attaching proximity sensors to the thumb [51], and using a microphone and gyroscope on the thumb [64]. However, these methods require additional sensors or devices, making it impossible to recognize gestures using only the smartwatch. Some studies have attempted to achieve thumb-to-finger gesture

[1] https://support.apple.com/en-gb/guide/watch/apdec70bfd2d/watchos
[2] https://www.polar.com/en/pacer
[3] https://www.suunto.com/Products/sports-watches/suunto-5/suunto-5-burgundy-copper/

MUM '24, December 01–04, 2024, Stockholm, Sweden

ACM ISBN 979-8-4007-1283-8/24/12
https://doi.org/10.1145/3701571.3701600

recognition using commercial off-the-shelf (COTS) smartwatch accelerometers [6, 23, 27, 55, 61] to eliminate the need for additional hardware. However, these studies have not specifically focused on thumb-to-finger gestures and have been limited to the recognition of only a subset of such gestures.

To fill these gaps, we aimed to achieve the recognition of a broader range of thumb-to-finger gestures using only COTS smartwatch's accelerometers. During a thumb-to-finger gesture, micro-vibrations, whose wave pattern is unique to each gesture, propagate from the hand to the smartwatch. Thus, gestures can be recognized by identifying their unique wave patterns. In a user study, we achieved 80.1% accuracy for 17 gestures. Moreover, by basing the optimal gesture set for different numbers of gestures on the accelerometer's recognition performance, we achieved 90.2% accuracy for 11 gestures and 98.2% accuracy for 5 gestures. We also investigated recognition performance under practical constraints, specifically limited training data, small window size, and low sampling rate. As a result, optimal gesture recognition is attained with a sampling rate between 400 Hz and 800 Hz and a window size of 1600 data points.

The study makes four contributions. First, we developed a method of thumb-to-finger gesture recognition using COTS smartwatch accelerometers and investigated its recognition performance. Second, we developed optimal gesture sets for each number of gestures based on the accelerometer's recognition performance. Third, we evaluated its recognition performance under practical constraints. Fourth, we demonstrated a potential application that enables eyes-free, one-handed, and rich input with no need for an additional sensor.

2 Related Work

Thumb-to-finger gestures have gained increasing attention in recent years, and accelerometer sensing, along with one-handed smartwatch control, has been widely explored in the HCI field. This section presents a review of the research on thumb-to-finger gestures, accelerometer sensing, and one-handed smartwatch input.

2.1 Thumb-to-finger Gestures

Researchers have explored design spaces, recognition methods, and applications for thumb-to-finger gestures.

2.1.1 Design Space. The design space of thumb-to-finger gestures has been comprehensively studied. FingerInput [48] identifies gestural primitives and consolidates them into a four-dimensional design space: (1) which finger touches, (2) which part of another finger is touched, (3) what touch action is performed, and (4) how fingers are flexed. Other studies have explored drawing various shapes, such as numbers [64], and have introduced tap force as an additional dimension of interaction [29]. Several studies have also focused on thumb-to-index-finger gestures, including interactions on the radial side of the index finger [3, 24, 53], pressure and hover distance [9], lateral thumb-index pinching [45], fingertip use [2, 13], and 2D fingertip input [21]. Moreover, some studies have examined the usability of gestures considering human factors [17, 20]. Research on user elicitation of single-hand micro-gestures has also been conducted [1].

2.1.2 Recognition Methods. Table 1 presents a comparison of our method with previously developed methods. Various devices and techniques for thumb-to-finger gesture recognition have been employed, including electrical impedance tomography [65], pressure sensors [7, 56], infrared transmission and reflection [36], and acoustic sensors [19, 25, 64]. Ring-based approaches [30, 51, 52, 54] and vision-based methods using wrist cameras [40], chest-mounted camera [34], and shoulder-mounted depth camera [48] have also been explored. Moreover, methods that track hand poses [8, 16, 28, 33, 58, 63] have been investigated.

Some studies have performed hand and thumb-to-finger gesture recognition using only the sensors built into smartwatches [6, 23, 27, 55, 61], but the range of thumb-to-finger gestures explored has been limited.

In contrast, our approach, which also requires no additional sensors, enables the recognition of 17 thumb-to-finger gestures. We also designed a gesture set taking into account the accelerometer's recognition performance.

2.1.3 Applications. Thumb-to-finger gestures have been successfully integrated into various applications, demonstrating their versatility and effectiveness in enhancing user interaction. For text entry, several keyboard layouts have been mapped onto finger segments, including the T9 layout on one hand [56], the QWERTY layout across both hands [40], an alphabetical layout on one hand [40], and the QWERTY layout precisely mapped onto the fingertips [62]. For computer manipulation, thumb-to-finger gestures have been used in multiple contexts. Thumb-to-finger tap gestures have been used for shortcut input in XR environments [40] and on wearable devices [34]. In XR, gestures such as thumb-to-radial-side-of-index-finger taps and swipes have also been used [24]. The commercially available Apple Watch incorporates AssistiveTouch, which enables smartwatch control using gestures such as pinch, double pinch, clench, and double clench.

2.2 Accelerometer Sensing

Research on gesture recognition using accelerometers has been conducted in various contexts.

2.2.1 Smartwatch Accelerometer. Several studies have explored gesture recognition using IMUs built into COTS smartwatches. For example, Serendipity [55] recognizes five finger gestures, such as pinching, tapping, and rubbing. ViBand [27] overclocks the accelerometer sampling rate of a COTS smartwatch to 4000 Hz, enabling the recognition of gestures such as flicking, clapping, scratching, and tapping. Taprint [4] recognizes tap locations on the back of the hand wearing a smartwatch by overclocking the IMU sampling rate. Xu et al. [61] developed a method for gesture customization based on a small amount of data without compromising the performance of existing gesture sets. Kimura [23] utilized self-supervised learning for hand gesture recognition using a small amount of labeled data. Chen et al. [6] improved the robustness of gesture recognition against variations in hand shape, finger strength, and smartwatch positioning by applying unsupervised Siamese adaptation. Other studies have used the IMUs to recognize screen tilts for interaction purposes, enabling target selection [14] and cursor control [26, 50] (available on AssistiveTouch). Accelerometers have

Table 1: Thumb-to-finger gesture recognition methods. The asterisk (∗) column indicates whether additional sensors are required (×) or not (✓). "Tap" indicates a tapping action. "Touch" refers to maintaining contact. "Slide" denotes a sliding motion between the thumb and the index finger. The dagger (†) denotes that thumb-to-finger gestures can alternatively be recognized using hand pose tracking.

Method	Sensor/technique (body part)	∗	Gestures
FingerT9 [56]	Pressure sensor (finger segments)	×	11 phalange taps
WristFlex [7]	Pressure sensor (wrist)	×	4 pinches
Tomo [65]	Impedance tomography (wrist)	×	4 pinches and clench
SensIR [36]	Infrared transmission and reflection (wrist)	×	5 pinches and clench
FingerInput [48]	Camera (head or shoulder)	×	Various gestures (e.g., taps, slides, circle)
PinchWatch [34]	Camera (chest)	×	4 pinches, slides, and circle
DigiTap [40]	Camera (wrist)	×	12 phalange taps
AudioTouch [25]	Speaker and microphone (back of the hand)	×	12 phalange touches
FingerSound [64]	Microphone and gyroscope (thumb)	×	4 directional swipes, 10 unistroke digits writing, and 28 graffiti letters writing
DualRing [30]	IMUs (thumb and index finger)	×	4 pinches and 12 phalange taps
ThumbRing [52]	Two IMUs (thumb and back of the hand)	×	10 phalange taps
OptiRing [54]	Camera (index finger)	×	Right swipe, left swipe, and long tap
ThumbTrak [51]	Proximity sensor (thumb)	×	12 phalange touches
Interferi [19]	Acoustic interferometry (wrist)	×	4 pinches and clench
[8, 16, 28, 33, 58, 63]	Camera, acoustic sensing or EMG (wrist)	×	Hand pose †
Serendipity [55]	COTS smartwatch accelerometer (wrist)	✓	1 pinch and finger rubbing
ViBand [27]	COTS smartwatch accelerometer (wrist)	✓	1 pinch, 1 flick, 1 snap, and finger rubbing
[6]	COTS smartwatch accelerometer (wrist)	✓	1 four-finger pinch, 1 slide, and clench
[23, 61]	COTS smartwatch accelerometer (wrist)	✓	index pinch, double index pinch, pinky pinch, double pinky pinch, slide, clench, and double clench
Our method	COTS smartwatch accelerometer (wrist)	✓	12 phalange taps, 4 pinches, and clench

also been employed to recognize characters drawn in the air or on a surface [5, 10, 32, 59]. Our approach focuses on recognizing more fine-grained thumb-to-finger gestures using only smartwatch accelerometers.

2.2.2 Non-Smartwatch Accelerometers. Research has also been conducted using external accelerometers. TapID and TapType [38, 49] identify which finger taps on a surface using accelerometers worn on both arms, with sampling rates of 1344 Hz and 1600 Hz, respectively. SparseIMU [46] recognizes fine-grained finger movements by fixing multiple IMUs to the arm and fingers. Smartphone accelerometers have also been used for such purposes. Studies [31, 37] have classified tap locations using neural networks, and TapNet [18] effectively recognizes one-handed interactions with smartphones.

2.3 One-handed Smartwatch Input

Several methods have been developed for one-handed smartwatch operation, including hand gestures [41, 58], elbow gestures [39], screen tilting for cursor control [26, 50] (available on AssistiveTouch), screen tilting for target selection [14], and wrist gestures [11, 12, 44]. Text input methods that recognize characters drawn with the hand wearing a smartwatch with no additional sensors have also been developed [5, 10, 32, 59]. However, these methods often have limitations, such as a limited input vocabulary, the need for additional sensors, or the inability to provide eyes-free interaction.

Thumb-to-finger gestures provide a richer input vocabulary and clear tactile feedback, which makes them suitable for fast, direct, and discreet eyes-free input with a single hand [17, 48]. These gestures can also extend the functionality of screen tilt-based cursor control methods. For instance, two different thumb-to-finger gestures can be used to select a target and open context menus. Several methods use thumb-to-finger gestures for one-handed smartwatch operation [17, 34, 56, 61]. PinchWatch [34] uses thumb-to-finger gestures recognized by a chest-mounted camera to control a smartwatch. DigitSpace [17] designs thumb-to-finger gestures for smartwatch operation, recognized through camera or magnetic sensing. FingerT9 [56] employs pressure sensors attached to finger segments to recognize thumb-to-finger gestures for T9 text input. The Apple Watch's AssistiveTouch feature uses accelerometer-recognized pinch and clench gestures for target selection.

Our approach aims to recognize a wider range of thumb-to-finger gestures using only COTS smartwatch accelerometers and to evaluate its recognition performance. We also propose a gesture set designed to align with the accelerometer's recognition performance.

3 Implementation

We used the built-in accelerometers on COTS smartwatches for thumb-to-finger gesture recognition without additional sensors. When a gesture is made, minute compressive waves propagate from the hand to the smartwatch. These waves vary depending on

Figure 1: Example of accelerometer-obtained x-axis gesture data.

the specific gesture (Figure 1). Leveraging this phenomenon, we developed a system that recognizes thumb-to-finger gestures using COTS smartwatch accelerometers. The system collects accelerometer data, detects gestures, extracts their time segments, and then classifies them using machine learning algorithms to distinguish the unique wave pattern associated with each gesture.

3.1 Apparatus

We used an LG G W100 smartwatch. For thumb-to-finger gesture recognition, we adopted an approach similar to that of ViBand [27], which increases the sampling rate to 4000 Hz, enabling the smartwatch to capture coarse motions and rich bio-acoustic signals. Thus, we collected accelerometer data at a sampling rate of 4000 Hz by modifying its Linux kernel. To account for real-time processing, we obtained accelerometer data from the smartwatch via a Bluetooth-paired smartphone and transferred them to a PC for signal processing and machine learning. To enhance performance, sensor data collection and subsequent signal processing were performed for every 50 samples (0.0125 seconds).

3.2 Gesture Detection and Time Segment Extraction

To classify thumb-to-finger gestures using machine learning, we detected them and extracted the corresponding segments from the accelerometer's time-series data. Gesture detection was based on identifying impacts in the data by calculating the rate of change in the signal. We employed the rate-of-change score R_x used in TapType [49] to analyze impacts across the three-axis accelerometer data. The R_x accumulates the absolute changes in magnitude across the three axes and combines them into a single dimension. To focus on gesture-specific changes and exclude low-frequency movements, such as arm motions, we applied a 15 Hz high-pass filter to each axis prior to calculating the R_x. To effectively capture subtle movements, we calculated the R_x over the previous 40 data points (0.01 seconds).

An overview of the gesture detection and time segment extraction is shown in Figure 2. Gestures were treated as discrete events within a fixed window. Detection was triggered when the maximum R_x value within the most recent window exceeded a threshold τ, and the gesture time segment was extracted when this maximum value was near the center of the window. Since the accelerometer data were processed in batches of 50 data points, we used a window size of 100 points to determine the window s center. This method enabled the extraction of gesture time segments from the start to the end of a gesture centered around its peak (the point with the highest R_x), preventing segment extraction in the middle of a gesture. To avoid multiple detections of the same gesture waveform, the detection was paused for the duration of the window size after each gesture was detected.

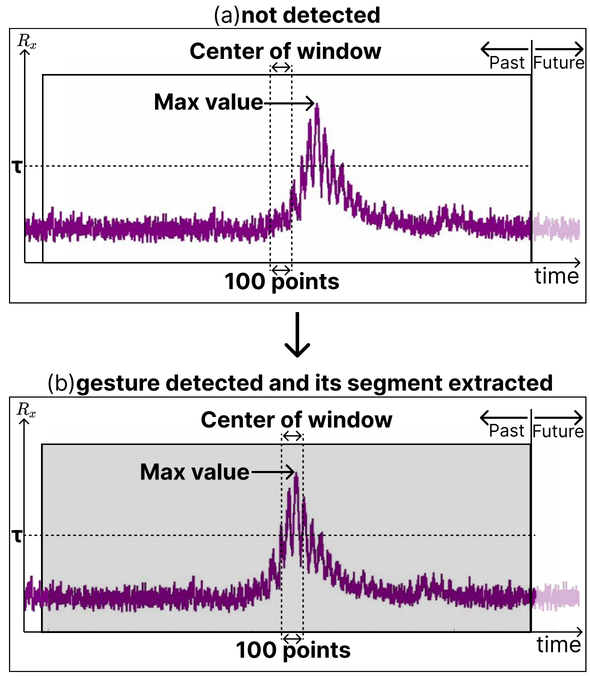

Figure 2: Overview of gesture detection and its time segment extraction. Detection is performed when the maximum value of R_x exceeds the threshold τ, and extraction is performed when it is located at the center of the window. (a) No gesture is detected. (b) A gesture is detected, and its time segment is extracted.

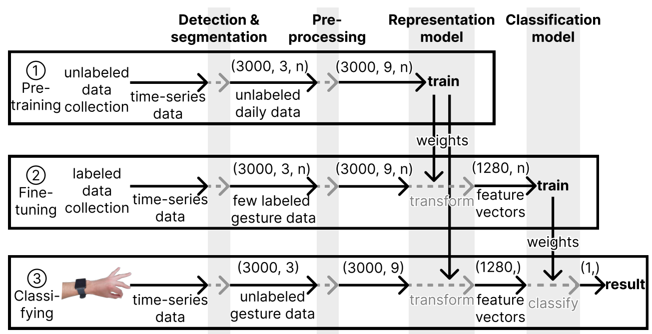

Figure 3: Overview of our classification system, inspired by [23].

3.3 Gesture Classification Methodology

We classified gestures based on the detected and extracted segments using machine learning. An overview of our classification system is shown in Figure 3. This system is based on Kimura's method [23], which uses self-supervised learning to enable few-shot hand gesture recognition, minimizing the need for labeled data for each user. The system comprises three stages: pre-training, fine-tuning, and classification.

3.3.1 Pre-Training (Unsupervised Representation Learning). The first stage involves pre-training through unsupervised representation learning. In this stage, we train a representation learning model to transform time-series data into feature vectors (i.e., representations). To collect training data, one of the authors performed daily activities while wearing a smartwatch for approximately two hours. Data were collected only when movement was detected based on an R_x threshold, resulting in 5106 data points, each with a 3000-long (0.75 seconds) window. We then divided the data into three frequency bands (0.22 Hz to 8 Hz, 8 Hz to 32 Hz, and above 32 Hz) following the previously developed procedure [23, 61]. The processed data were then used to train the representation learning model.

3.3.2 Fine-Tuning (Supervised Learning). The second stage is fine-tuning through supervised learning. In this stage, we train the classification model using a small amount of labeled user gesture data. For this purpose, we used a linear support vector machine for the classification model. We divided accelerometer data into the same three frequency bands as in the pre-training stage. Using the pre-trained representation learning model, we transformed the time-series data into 1280-dimensional feature vectors. Unlike a previous study [23] that used 360 dimensions, we used 1280 dimensions to accommodate a larger volume of time-series data (3000 data points). We then used transformed data to train the gesture classification model.

3.3.3 Classification. The third stage is the classification of gestures. Here, we classified the unlabeled gesture data performed by the user. First, we divided the user-performed gesture data into the same three frequency bands as in the previous stages. We then used the pre-trained representation learning model to transform the time-series data into 1280-dimensional feature vectors. Finally, we used the fine-tuned classification model (linear support vector machine) to classify the gestures.

4 User Study

We conducted a user study to evaluate the thumb-to-finger gesture recognition performance using COTS smartwatch accelerometers. For this purpose, we recruited 12 participants (10 males, 2 females) aged 22-25 years (average: 23.2 years) who volunteered from our laboratory. All participants were right-handed. The study took approximately 30 minutes on average.

4.1 Design

We designed the gesture set and user study.

4.1.1 Gesture Set. We designed a set of 17 gestures (Figure 4) based on two considerations, same as [60]. To ensure that the gestures would be easy and natural to remember for most users, we avoided gestures involving more than two fingers. Moreover, we selected subtle and fine-grained gestures [1] that required minimal effort. Specifically, we adopted three types of gestures: tap [48], flick [60] and clench. Since each gesture consisted of a sequence of finger movements, we clearly defined the start and end positions of the

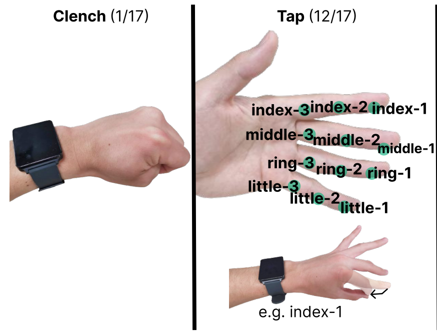

Figure 4: The designed gestures.

fingers for data labeling, as described in [60]. These definitions were essential for ensuring consistent gesture labeling across participants and sessions. The tap gestures began with the thumb near the target area, followed by a tap on the target area, and ended by returning to the initial position. The flick gestures began with the fingers positioned close to the thumb, followed by a quick flicking motion by extending the fingers, and ended with the fingers extended. The clench gestures started in a relaxed (neutral) state, followed by clenching, and ended by returning to the initial position. For the tap gestures, we divided each finger into three sections at the joints, resulting in 12 tap targets (Figure 4): index-1, index-2, index-3, middle-1, middle-2, middle-3, ring-1, ring-2, ring-3, little-1, little-2, and little-3. For the flick gestures, we defined four targets (Figure 4): index-flick, middle-flick, ring-flick, and little-flick. Thus, we defined a total of 17 gestures.

4.1.2 Gesture Recognition Performance Assessment. We performed gesture detection and time segmentation as described in the previous section. Since each gesture lasted approximately 0.75 seconds, we set the window size to 3000 data points (4000 points per second $\times$ 0.75 seconds).

We performed leave-one-participant-out cross-validation to assess inter-participant recognition performance and leave-one-session-out cross-validation to assess participant-specific recognition performance when the participants put the smartwatch back on for a new session. To account for the challenges of recognizing a wide variety of gestures with high accuracy, we also evaluated recognition performance specifically when focusing on a subset of gestures selected based on the recognition performance of the accelerometer. Moreover, we assessed recognition performance with reduced training data to minimize calibration efforts in practical use. We also evaluated how truncating the latter part of the gesture waveform for quicker recognition affected performance. Finally, we assessed recognition performance at lower sampling rates to address power consumption concerns.

4.2 Task and Instructions

The participants wore the smartwatch on the wrist on which they usually wore a watch (the left wrist in all cases). To reduce fatigue during the experiment, the participants sat in a chair with their left elbow resting on a desk, ensuring the smartwatch did not touch the desk. They then performed gestures with their left hand while

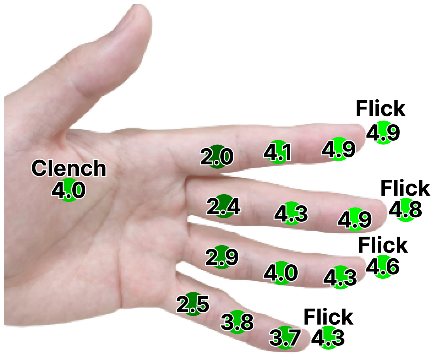

Figure 5: Gesture comfortableness scores. Scores without labels pertain to tap gestures. A score of 5 indicates "very comfortable," while a score of 1 indicates "very uncomfortable."

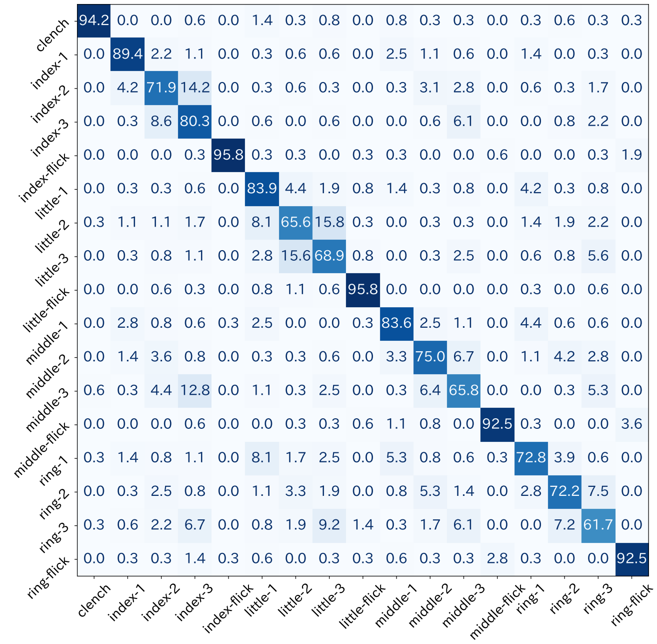

Figure 6: Confusion matrix of participant-specific leave-one-session-out cross-validation averaged across all participants. The values are in parentages.

looking at the smartwatch screen, which displayed the name of the gesture to be performed. We instructed the participants to perform consistent finger movements for the same gesture, maintain a steady body and arm posture, and execute the gestures quickly (within 0.75 seconds) to ensure the gesture waveform was fully captured within the window.

4.3 Procedure

The participants performed one practice session and six main sessions. The aim of the practice session was to help the participants familiarize themselves with the gestures. In each session, they performed each gesture five times consecutively, with the order of gestures remaining the same across sessions. The participants had a break of at least 1 minute between sessions, during which they removed the smartwatch. After completing the sessions, they completed a questionnaire in which they rated the comfortableness of each gesture on a five-point Likert scale (1: very uncomfortable, 5: very comfortable). The total duration of the experiment for each participant was approximately 30 minutes.

5 Results and Analysis

We collected 510 samples per participant, calculated as 17 types of gestures $\times$ 5 repetitions $\times$ 6 sessions. This section presents the results of the user study and provides an analysis.

5.1 Comfortableness

We treated the five-point Likert scale as an interval scale where "very comfortable" was rated as 5, and "very uncomfortable" was rated as 1. The average comfortableness score for each gesture is shown in Figure 5. Gestures involving fingers farther from the thumb or inner finger segments were rated less comfortable, likely due to the thumb's anatomy. This is consistent with previous studies on tap gesture comfortableness [17, 20].

5.2 Recognition of All Gestures

We first investigated recognition performance across all gestures. We conducted leave-one-participant-out cross-validation to evaluate how well a model trained on other participants' data recognizes the gestures of a new participant. Moreover, we performed participant-specific leave-one-session-out cross-validation to evaluate the need for recalibration after putting the device back on when participant-specific calibration was applied.

5.2.1 Leave-One-Participant-Out Cross-Validation. The overall gesture classification accuracy was 38.8%. We observed frequent misclassification of gestures performed with the same finger and those targeting the same part of a finger (tip, middle, or base).

5.2.2 Participant-Specific Leave-One-Session-Out Cross-Validation. The overall classification accuracy was 80.1% (SD: 9.32%). A confusion matrix is shown in Figure 6. We observed frequent misclassification of gestures performed with the same finger and those targeting the same part of a finger (tip, middle, or base). Moreover, recognition performance varied among participants, which may be due to differences in how consistently they repeated the same gestures with similar finger movements.

5.3 Exploration of the Optimal Gesture Set for Each Number of Gestures

Given the difficulty of recognizing all gestures with high accuracy in both cross-validation types, we explored the optimal gesture set for each number of gestures. We selected gestures based on the recognition performance of accelerometer sensing. Following previous research [17, 20], we excluded four gestures (index-3, middle-3,

Table 2: Optimal gesture set for each number of gestures in leave-one-participant-out cross-validation. The leftmost column represents the number of gestures, and the rightmost column represents the accuracy in %.

#	Target gesture													ACC
13	index-1	middle-1	ring-1	little-1	index-2	middle-2	ring-2	little-2	index-flick	middle-flick	ring-flick	little-flick	clench	47.3
12	index-1	middle-1		little-1	index-2	middle-2	ring-2	little-2	index-flick	middle-flick	ring-flick	little-flick	clench	50.9
11	index-1	middle-1		little-1	index-2	middle-2	ring-2	little-2	index-flick		ring-flick	little-flick	clench	55.4
10	index-1	middle-1		little-1	index-2		ring-2	little-2	index-flick		ring-flick	little-flick	clench	60.1
9		middle-1		little-1	index-2		ring-2	little-2	index-flick		ring-flick	little-flick	clench	65.0
8		middle-1			index-2		ring-2	little-2	index-flick		ring-flick	little-flick	clench	70.0
7		middle-1			index-2			little-2	index-flick		ring-flick	little-flick	clench	76.5
6		middle-1						little-2	index-flick		ring-flick	little-flick	clench	83.9
5		middle-1						little-2	index-flick			little-flick	clench	89.0
4		middle-1							index-flick			little-flick	clench	94.7
3									index-flick			little-flick	clench	98.1
2									index-flick				clench	99.6

Table 3: Optimal gesture set for each number of gestures in participant-specific leave-one-session-out cross-validation. The leftmost column represents the number of gestures, and the rightmost column represents the accuracy in %.

#	Target gesture													ACC
13	index-1	middle-1	ring-1	little-1	index-2	middle-2	ring-2	little-2	index-flick	middle-flick	ring-flick	little-flick	clench	86.7
12	index-1	middle-1		little-1	index-2	middle-2	ring-2	little-2	index-flick	middle-flick	ring-flick	little-flick	clench	88.8
11	index-1	middle-1		little-1	index-2	middle-2	ring-2		index-flick	middle-flick	ring-flick	little-flick	clench	90.2
10	index-1	middle-1		little-1	index-2		ring-2		index-flick	middle-flick	ring-flick	little-flick	clench	92.1
9	index-1	middle-1			index-2		ring-2		index-flick	middle-flick	ring-flick	little-flick	clench	93.2
8	index-1	middle-1					ring-2		index-flick	middle-flick	ring-flick	little-flick	clench	94.9
7	index-1						ring-2		index-flick	middle-flick	ring-flick	little-flick	clench	96.6
6	index-1						ring-2		index-flick	middle-flick		little-flick	clench	97.6
5	index-1								index-flick	middle-flick		little-flick	clench	98.2
4									index-flick	middle-flick		little-flick	clench	98.8
3									index-flick			little-flick	clench	99.4
2												little-flick	clench	99.7

ring-3, and little-3) that had an average comfortableness rating of less than 3 to create a gesture set that considers user comfort. To identify the optimal gesture set for each number of gestures while considering the recognition performance of the accelerometer, we repeatedly performed the following steps on the remaining gestures:

(1) Performing cross-validation on the remaining gestures and computing the F1 score for each gesture.
(2) Excluding the gesture with the lowest F1 score.

We used the F1 score because it balances both precision and recall. This approach results in the exclusion of one gesture from each set of similar gestures. The number of gestures, their corresponding gesture sets, and the accuracy obtained are shown in Table 2 and Table 3. Accuracy exceeded 90% for 4 or fewer gestures in leave-one-participant-out cross-validation and for 11 or fewer gestures in participant-specific leave-one-session-out cross-validation.

5.4 Evaluation of Performance Under Practical Constraints

Since participant-specific leave-one-session-out cross-validation yielded higher accuracy, we further investigated recognition performance under practical constraints by reducing the training data, window size, and sampling rate. For these evaluations, we used sets of 5, 11, and 17 gestures, as determined by the gesture elimination process; the specific gesture types are detailed in Table 3.

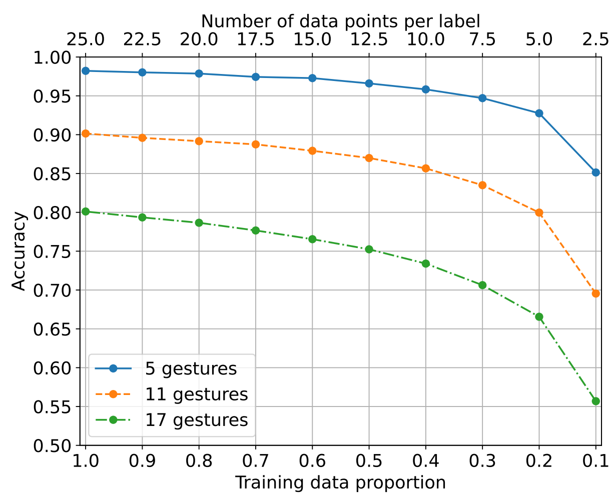

Figure 7: Accuracy when the training data size in participant-specific leave-one-session-out cross-validation was reduced from 1.0 to 0.1.

5.4.1 Reducing the Training Data. To reduce calibration times for practical use, we examined the effect of reducing the size of training data while maintaining acceptable accuracy. We randomly sampled

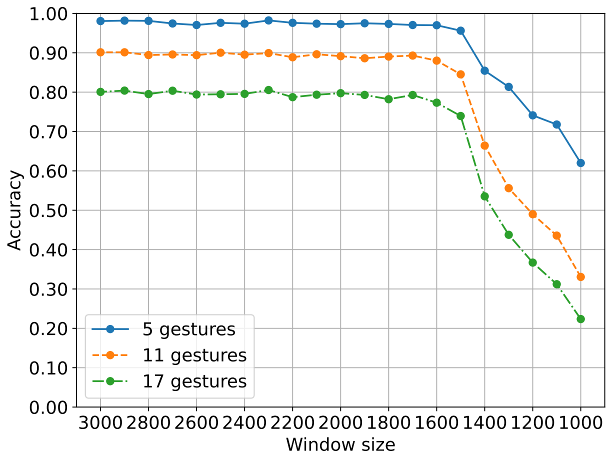

Figure 8: Accuracy when the window size in participant-specific leave-one-session-out cross-validation was reduced from 3000 to 1000.

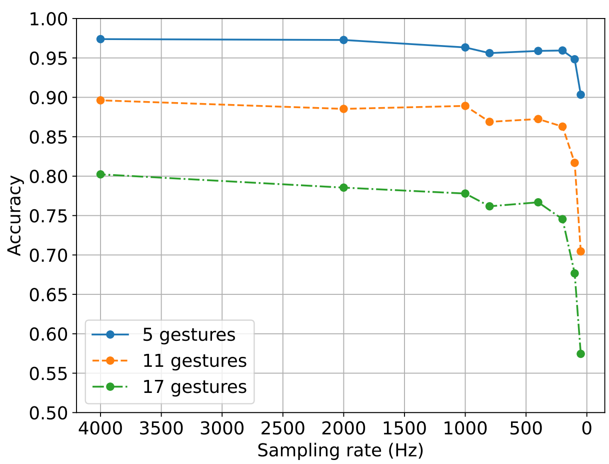

Figure 9: Accuracy when the sampling rate in participant-specific leave-one-session-out cross-validation was reduced from 4000 Hz to 50 Hz.

a specified proportion of the training data, ensuring an even representation of gesture types to avoid bias. To account for randomness, we repeated this process 10 times and averaged the results. As shown in Figure 7, accuracy decreased monotonically as the training data were reduced. Moreover, the rate of improvement diminished with increased data, indicating a limit to the benefit of adding more data.

5.4.2 Reducing the Window Size. To improve recognition speed, we explored using only the initial portion of the gesture waveform by truncating the latter part of each data window while maintaining accuracy, as we had also done during pre-training. As shown in Figure 8, larger window sizes generally led to higher accuracy, with minimal differences, until the window size was reduced to 1600.

5.4.3 Reducing the Sampling Rate. To reduce power consumption, we examined the effect of lowering the sampling rate while maintaining acceptable accuracy. We retained only every second, third, or nth data point from the original sequence, as we had also done during pre-training. As shown in Figure 9, higher sampling rates generally resulted in higher accuracy, with accuracy decreasing gradually as the rate decreased to 200 Hz and then sharply beyond that point.

6 Discussion

In the results section, we presented the overall recognition accuracy, the exploration of the optimal gesture set for each number of gestures, and evaluation of performance under practical constraints. In this section, we discuss the gesture recognition performance results, potential applications, and limitations.

6.1 Gesture Recognition

The leave-one-participant-out cross-validation results suggest that recognizing a large number of gestures with high accuracy is challenging. This difficulty is likely due to variations in the way in which individuals perform gestures, which are influenced by personal habits and differences in movement. We observed frequent misclassification of gestures performed with the same finger or targeting the same part of a finger. To address this issue, we explored the optimal gesture set for each number of gestures, considering the recognition performance of accelerometer sensing. This approach led to recognition accuracies of 94.7% for four gestures, 98.1% for three gestures, and 99.6% for two gestures. This method allows gesture recognition without user-specific calibration using a model trained on data from other users.

In the participant-specific leave-one-session-out cross-validation, we achieved an overall recognition accuracy of 80.1%. As in the previous validation, we observed frequent misclassification of gestures involving the same finger or targeting the same part of a finger. By optimizing the gesture set while considering the accelerometer's recognition performance, we achieved recognition accuracies of 90.2% for 11 gestures and 98.2% for 5 gestures. Achieving these accuracies required collecting 25 data samples per gesture for each user. However, even with just 10 samples per gesture, we achieved recognition accuracies of 85.7% for 11 gestures and 95.8% for 5 gestures.

Prior studies on single-handed (not necessarily thumb-to-finger) gesture recognition using smartwatch accelerometers have achieved accuracies of 94.4% for 6 types of hand gestures [6], 87.0% for 5 types of hand gestures [55], approximately 94.3% (details not specified) for 6 types of hand gestures [27], 95.7% for 4 types of thumb-to-finger gestures and 87.2% for 16 types of hand gestures [61], and 95.0% for 10 types of hand gestures [23]. Although a direct comparison cannot be made due to differences in the amount of data collected, target gestures, participants, and recognition models, we achieved performance comparable to or better than that achieved in these studies.

When reducing the window size, we observed minimal changes in accuracy until the window size decreased to 1600. With the gesture detection and segmentation system used in this study, a window size of 1600 enables gesture recognition 0.35 seconds faster

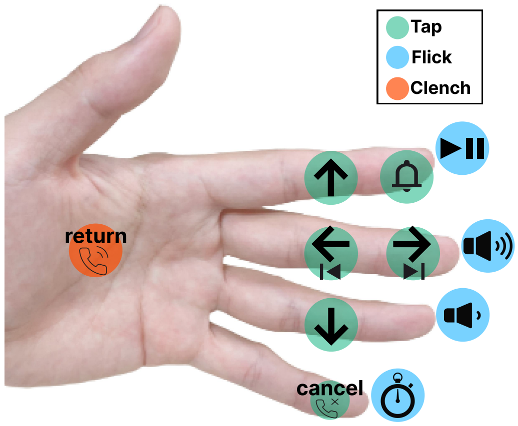

Figure 10: Example application.

than achieved with a window size of 3000. While a window size of 1600 appears to be more advantageous, the trade-off between recognition speed and accuracy should be evaluated based on user preferences.

Commercial smartwatches typically offer sampling rates between 50 Hz and 200 Hz through their APIs[4], with the Apple Watch providing a sampling rate of up to 800 Hz[5]. While higher sampling rates are generally preferable, a rate between 400 Hz and 800 Hz is ideal for thumb-to-finger gesture recognition that balances accuracy and power consumption.

6.2 Applications

Thumb-to-finger gestures require only one hand, which makes them particularly useful when the other hand is occupied (e.g., carrying items, cooking, or holding an umbrella) or for users with limited use of one hand. They are also suitable for smartwatches without a touchscreen, such as sports wristbands, and for cases in which the screen is already in use, such as when a video is being played, leaving no space for on-screen controls. Moreover, such gestures provide flexibility in various tasks. For instance, with a small set of two to five gestures, users can perform functions such as adjusting the volume, navigating ("back", "forward", "confirm"), entering text using the H4-Writer algorithm [35] and executing D-pad operations ("up", "down", "left", "right", "confirm").

A larger set of gestures enables more complex operations and a richer input vocabulary. For example, our set of 11 thumb-to-finger gestures, which were recognized with 90.2% accuracy, can provide sufficient versatility for a range of commands. For shortcut operations, our set of seven gestures could be used for common functions such as playing/resuming, increasing/decreasing the volume, starting/stopping a stopwatch, showing notifications, returning (e.g., answering calls), and canceling (e.g., declining calls). This flexibility can also enable the integration of D-pad controls with shortcut operations (see Figure 10). For text input, a more extensive set of gestures can increase the number of available keys and reduce word prediction conflicts when using word prediction.

Because thumb-to-finger gestures can be performed without looking ("eyes-free"), they are also well-suited for controlling other devices, such as smartphones, tablets, laptops, and XR (i.e., where users are wearing a head-mounted display (HMD)). Users are not required to keep their hands within the HMD camera's field of view, allowing them to keep their arms in a relaxed, lowered position [17, 48]. For example, users can quickly pause a video, navigate using the D-pad, or control music playback while keeping their arms lowered. Moreover, they can use gestures to control smart home devices, such as lights or air conditioners.

6.3 Limitations

Our study has several limitations. First, the tilt of the arm and the way in which gestures are performed may vary due to individual habits in practical use. This variability could be addressed via Siamese adaptation [6]. Moreover, the user study was conducted with the participants in a sitting position. Because disturbances caused by walking may interfere with accelerometer signals, further research is needed to assess recognition performance while the user is walking. False gesture detections from everyday movements, such as walking, could be mitigated by checking if the screen is facing upward, as it is unlikely to be in this orientation during walking.

Second, we focused on three types of gestures: tap, flick, and clench. FingerInput [48] defines a broader range of thumb-to-finger gestures, such as sliding the thumb across a finger or drawing a circle on the fingers with the thumb. Incorporating these gestures may enable more diverse operations. Thus, future research could explore the accelerometer's recognition performance of these gestures.

Third, while we developed the gesture set by considering the accelerometer's recognition performance, this approach may result in inconsistent gesture sets. For example, gestures such as index-1, middle-1, and little-1 are included, whereas ring-1 is omitted. Therefore, it is necessary to evaluate the usability of the selected gesture set and design gesture sets with usability in mind.

Fourth, the user study involved only 12 participants, all of whom belonged to the same age group. To examine variations in recognition performance due to differences in the ways in which individuals perform gestures according to personal habits, it is necessary to conduct a larger experiment involving a greater number of participants from more age groups.

Finally, we implemented the recognition system on a PC. For practical use, the system should be executable on a smartwatch itself or a connected smartphone. However, we did not examine whether the developed system can operate on these devices within acceptable computational costs. Future work should focus on developing a recognition system by taking into account the computational capabilities of such devices.

7 Conclusion

In this paper, we examined thumb-to-finger gesture recognition using a COTS smartwatch accelerometer. Our recognition system had an overall accuracy of 80.1% accuracy for 17 gestures. By considering the performance of accelerometer-based recognition, we

[4]https://source.android.com/docs/compatibility/14/android-14-cdd#731_accelerometer
[5]https://developer.apple.com/videos/play/wwdc2023/10179/

achieved 94.7% accuracy for 4 gestures in leave-one-participant-out cross-validation and 90.2% accuracy for 11 gestures in participant-specific leave-one-session-out cross-validation. Our analysis indicates that optimal gesture recognition is attained with a sampling rate between 400 Hz and 800 Hz and a window size of 1600 data points (0.4 seconds). This approach enables one-handed smartwatch control without additional sensors, providing eyes-free and rich input.

References

[1] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-Hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3403–3414.

[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* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 255–260.

[3] Taizhou Chen, Tianpei Li, Xingyu Yang, and Kening Zhu. 2023. EFRing: Enabling Thumb-to-Index-Finger Microgesture Interaction through Electric Field Sensing Using Single Smart Ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 161 (jan 2023), 31 pages.

[4] Wenqiang Chen, Lin Chen, Yandao Huang, Xinyu Zhang, Lu Wang, Rukhsana Ruby, and Kaishun Wu. 2019. Taprint: Secure Text Input for Commodity Smart Wristbands. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 17, 16 pages.

[5] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1, Article 45 (2021), 25 pages.

[6] Wenqiang Chen, Ziqi Wang, Pengrui Quan, Zhencan Peng, Shupei Lin, Mani Srivastava, Wojciech Matusik, and John Stankovic. 2023. Robust Finger Interactions with COTS Smartwatches via Unsupervised Siamese Adaptation. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 25, 14 pages.

[7] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 161–166.

[8] Nathan Devrio and Chris Harrison. 2022. DiscoBand: Multiview Depth-Sensing Smartwatch Strap for Hand, Body and Environment Tracking. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 56, 13 pages.

[9] Rhett Dobinson, Marc Teyssier, Jürgen Steimle, and Bruno Fruchard. 2022. MicroPress: Detecting Pressure and Hover Distance in Thumb-to-Finger Interactions. In *Proceedings of the 2022 ACM Symposium on Spatial User Interaction* (Online, CA, USA) *(SUI '22)*. Association for Computing Machinery, New York, NY, USA, Article 4, 10 pages.

[10] Yi Gao, Siyu Zeng, Ji Zhao, Wenxin Liu, and Wei Dong. 2021. AirText: One-Handed Text Entry in the Air for COTS Smartwatches. *IEEE Transactions on Mobile Computing* 22, 5 (2021), 2506–2519.

[11] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang 'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. WrisText: One-Handed Text Entry on Smartwatch Using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 181, 14 pages.

[12] Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. WristWhirl: One-Handed Continuous Smartwatch Input Using Wrist Gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 861–872.

[13] Jun Gong, Yang Zhang, Xia Zhou, and Xing-Dong Yang. 2017. Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 553–563.

[14] Anhong Guo and Tim Paek. 2016. Exploring Tilt for No-Touch, Wrist-Only Interactions on Smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 17–28.

[15] Teresa Hirzle, Jan Rixen, Jan Gugenheimer, and Enrico Rukzio. 2018. WatchVR: Exploring the Usage of a Smartwatch for Interaction in Mobile Virtual Reality. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6.

[16] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 71 (jun 2020), 24 pages.

[17] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1526–1537.

[18] Michael Xuelin Huang, Yang Li, Nazneen Nazneen, Alexander Chao, and Shumin Zhai. 2021. TapNet: The Design, Training, Implementation, and Applications of a Multi-Task Learning CNN for Off-Screen Mobile Input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 282, 11 pages.

[19] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture Sensing using On-Body Acoustic Interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13.

[20] 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).

[21] Keiko Katsuragawa, Ju Wang, Ziyang Shan, Ningshan Ouyang, Omid Abari, and Daniel Vogel. 2019. Tip-Tap: Battery-free Discrete 2D Fingertip Input. 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, 1045–1057.

[22] Daniel Kharlamov, Brandon Woodard, Liudmila Tahai, and Krzysztof Pietroszek. 2016. TickTockRay: smartwatch-based 3D pointing for smartphone-based virtual reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (Munich, Germany) *(VRST '16)*. Association for Computing Machinery, New York, NY, USA, 365–366.

[23] Naoki Kimura. 2022. Self-Supervised Approach for Few-Shot Hand Gesture Recognition. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 21, 4 pages.

[24] Kenrick Kin, Chengde Wan, Ken Koh, Andrei Marin, Necati Cihan Camgöz, Yubo Zhang, Yujun Cai, Fedor Kovalev, Moshe Ben-Zacharia, Shannon Hoople, Marcos Nunes-Ueno, Mariel Sanchez-Rodriguez, Ayush Bhargava, Robert Wang, Eric Sauser, and Shugao Ma. 2024. STMG: A Machine Learning Microgesture Recognition System for Supporting Thumb-Based VR/AR Input. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 753, 15 pages.

[25] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. 2019. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 36, 13 pages.

[26] Hiroki Kurosawa, Daisuke Sakamoto, and Tetsuo Ono. 2018. MyoTilt: A Target Selection Method for Smartwatches Using the Tilting Operation and Electromyography. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Barcelona, Spain) *(MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, Article 43, 11 pages.

[27] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 321–333.

[28] Chi-Jung Lee, Ruidong Zhang, Devansh Agarwal, Tianhong Catherine Yu, Vipin Gunda, Oliver Lopez, James Kim, Sicheng Yin, Boao Dong, Ke Li, Mose Sakashita, Francois Guimbretiere, and Cheng Zhang. 2024. EchoWrist: Continuous Hand Pose Tracking and Hand-Object Interaction Recognition Using Low-Power Active Acoustic Sensing On a Wristband. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 403, 21 pages.

[29] Lik Hang Lee, Kit Yung Lam, Tong Li, Tristan Braud, Xiang Su, and Pan Hui. 2019. Quadmetric Optimized Thumb-to-Finger Interaction for Force Assisted One-Handed Text Entry on Mobile Headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 94 (sep 2019), 27 pages.

[30] 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.

[31] Yi Liang, Zhipeng Cai, Jiguo Yu, Qilong Han, and Yingshu Li. 2018. Deep learning based inference of private information using embedded sensors in smart devices. *IEEE Network* 32, 4 (2018), 8–14.

[32] Xinye Lin, Yixin Chen, Xiao-Wen Chang, Xue Liu, and Xiaodong Wang. 2018. SHOW: Smart Handwriting on Watches. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4, Article 151 (2018), 23 pages.

[33] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1471–1482.

[34] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: A Wearable Device for One-Handed Microinteractions. In *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services Workshop on Ensembles of On-Body Devices (MobileHCI '10)*. Association for Computing Machinery, New York, NY, USA.

[35] I. Scott MacKenzie, R. William Soukoreff, and Joanna Helga. 2011. 1 Thumb, 4 Buttons, 20 Words Per Minute: Design and Evaluation of H4-Writer. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 471–480.

[36] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017. SensIR: Detecting Hand Gestures with a Wearable Bracelet using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 593–597.

[37] Maryam Mehrnezhad, Ehsan Toreini, Siamak F Shahandashti, and Feng Hao. 2018. Stealing PINs via mobile sensors: actual risk versus user perception. *International Journal of Information Security* 17, 3 (2018), 291–313.

[38] 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.

[39] Florian Müller, Sebastian Günther, Niloofar Dezfuli, Mohammadreza Khalilbeigi, and Max Mühlhäuser. 2016. ProxiWatch: Enhancing Smartwatch Interaction through Proximity-based Hand Input. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 2617–2624.

[40] 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)*. Association for Computing Machinery, New York, NY, USA, 9–18.

[41] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. 2018. SynchroWatch: One-Handed Synchronous Smartwatch Gestures Using Correlation and Magnetic Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4, Article 158 (2018), 26 pages.

[42] Julie Rico and Stephen Brewster. 2010. Usable gestures for mobile interfaces: evaluating social acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 887–896. https://doi.org/10.1145/1753326.1753458

[43] Franca Alexandra Rupprecht, Achim Ebert, Andreas Schneider, and Bernd Hamann. 2017. Virtual Reality Meets Smartwatch: Intuitive, Natural, and Multi-Modal Interaction. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 2884–2890.

[44] Farshid Salemi Parizi, Wolf Kienzle, Eric Whitmire, Aakar Gupta, and Hrvoje Benko. 2021. RotoWrist: Continuous Infrared Wrist Angle Tracking using a Wristband. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology* (Osaka, Japan) *(VRST '21)*. Association for Computing Machinery, New York, NY, USA, Article 26, 11 pages.

[45] Martin Schmitz, Sebastian Günther, Dominik Schön, and Florian Müller. 2022. Squeezy-Feely: Investigating Lateral Thumb-Index Pinching as an Input Modality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 61, 15 pages.

[46] Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. 2023. SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-grained Finger Microgestures. *ACM Trans. Comput.-Hum. Interact.* 30, 3, Article 39 (jun 2023), 40 pages.

[47] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (Rome, Italy) *(INTERACT '05)*. Springer-Verlag, Berlin, Heidelberg, 267–280.

[48] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (Tokyo, Japan) *(ISS '18)*. Association for Computing Machinery, New York, NY, USA, 177–187.

[49] 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 *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 497, 16 pages.

[50] Ke Sun, Yuntao Wang, Chun Yu, Yukang Yan, Hongyi Wen, and Yuanchun Shi. 2017. Float: One-Handed and Touch-Free Target Selection on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 692–704.

[51] Wei Sun, Franklin Mingzhe Li, Congshu Huang, Zhenyu Lei, Benjamin Steeper, Songyun Tao, Feng Tian, and Cheng Zhang. 2021. ThumbTrak: Recognizing Micro-finger Poses Using a Ring with Proximity Sensing. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction* (Toulouse & Virtual, France) *(MobileHCI '21)*. Association for Computing Machinery, New York, NY, USA, Article 2, 9 pages.

[52] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. 2016. ThumbRing: private interactions using one-handed thumb motion input on finger segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 791–798.

[53] Hsin-Ruey Tsai, Te-Yen Wu, Da-Yuan Huang, Min-Chieh Hsiu, Jui-Chun Hsiao, Yi-Ping Hung, Mike Y. Chen, and Bing-Yu Chen. 2017. SegTouch: Enhancing Touch Input While Providing Touch Gestures on Screens Using Thumb-To-Index-Finger Gestures. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 2164–2171.

[54] Anandghan Waghmare, Roger Boldu, Eric Whitmire, and Wolf Kienzle. 2023. OptiRing: Low-Resolution Optical Sensing for Subtle Thumb-to-Index Micro-Interactions. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction* (Sydney, NSW, Australia) *(SUI '23)*. Association for Computing Machinery, New York, NY, USA, Article 8, 13 pages.

[55] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3847–3851.

[56] 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)*. Association for Computing Machinery, New York, NY, USA, Article 178, 10 pages.

[57] Pui Chung Wong, Kening Zhu, Xing-Dong Yang, and Hongbo Fu. 2020. Exploring Eyes-free Bezel-initiated Swipe on Round Smartwatches. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11.

[58] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M. Kitani. 2020. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-Worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1147–1160.

[59] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-Writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (Santa Fe, New Mexico, USA) *(HotMobile '15)*. Association for Computing Machinery, New York, NY, USA, 9–14.

[60] Chenhan Xu, Bing Zhou, Gurunandan Krishnan, and Shree Nayar. 2023. AO-Finger: Hands-Free Fine-Grained Finger Gesture Recognition via Acoustic-Optic Sensor Fusing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23, 306)*. Association for Computing Machinery, New York, NY, USA, 14 pages.

[61] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery,

New York, NY, USA, Article 496, 19 pages.

[62] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 883–899.

[63] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. 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, 963–971. https://doi.org/10.1145/3332165.3347867

[64] 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.

[65] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 167–173.