ThumbTrak: Recognizing Micro-finger Poses Using a Ring with Proximity Sensing

Wei Sun*
Cornell University
United States
Institute of Software, Chinese
Academy of Sciences
China

Franklin Mingzhe Li
Carnegie Mellon University
United States

Congshu Huang
Cornell University
United States
Peking University
China

Zhenyu Lei
Cornell University
United States
Huazhong University of Science and
Technology
China

Benjamin Steeper
Cornell University
United States

Songyun Tao
Cornell University
United States

Feng Tian[†]
School of Artificial Intelligence,
University of Chinese Academy of
Sciences, Beijing
China

Cheng Zhang[‡]
Cornell University
United States

ABSTRACT

ThumbTrak is a novel wearable input device that recognizes 12 micro-finger poses in real-time. Poses are characterized by the thumb touching each of the 12 phalanges on the hand. It uses a thumb-ring, built with a flexible printed circuit board, which hosts nine proximity sensors. Each sensor measures the distance from the thumb to various parts of the palm or other fingers. ThumbTrak uses a support-vector-machine (SVM) model to classify finger poses based on distance measurements in real-time. A user study with ten participants showed that ThumbTrak could recognize 12 micro finger poses with an average accuracy of 93.6%. We also discuss potential opportunities and challenges in applying ThumbTrak in real-world applications.

CCS CONCEPTS

• **Human-centered computing → Pointing devices**.

*Also with School of Computer Science and Technology, University of Chinese Academy of Sciences, China.
†Also with State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Sciences, China.
‡Corresponding author

KEYWORDS

sensing, wearable sensing, gesture recognition, micro-finger poses, proximity sensing, machine learning

1 INTRODUCTION

The state-of-art interaction techniques on wearable computers (e.g., smartwatch, glass) are dominated by speech-based and touchscreen-based interaction, which have not yet provided the optimal interaction experience for certain scenarios. Speech interaction may not work well in public spaces, where environmental noise may be high or talking out loud might be socially inappropriate [22]. Touch-based interfaces on wearables are often limited by the screen sizes [27]. Therefore, the expressiveness of the touch interaction on wearables is relatively limited, compared to the touch interaction on tablets or phones. Furthermore, touch-based interaction demands exclusive eye-contact, which may not always be convenient. For instance, if a phone call comes in during a meeting, the user may want to immediately dismiss the phone call, instead of locating the 'reject' button on a screen or issue voice commands [29]. Therefore, there is a need for a discrete and eyes-free input technique to promote wearable interaction experience in these scenarios.

Existing work has explored various approaches to enable hand or finger gestures for wearable devices (e.g., [17, 25]). Many of them rely on wearing sensor units on the wrist or forearm to recognize
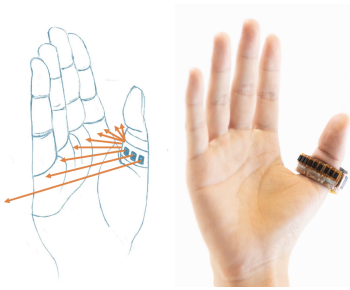
Figure 1: ThumbTrak System

Figure 2: ThumbTrak Pose Family

finger or hand gestures (e.g., [10, 17, 36]). However, such methods either require heavy instrumentation, which might be obtrusive to the user, or the recognition performance from sensor units on the wrist or forearm might be affected in certain situations, such as the occlusions from clothing or certain hand angles. Therefore, we are interested in exploring ring-based approaches because many people are used to wearing rings in everyday activities and it has potentials to detect various hand/finger gestures with less constraints (e.g., hand angles or clothing conditions) than prior approaches.

In our work, we present ThumbTrak (as shown in Figure 1), a thumb-ring based eyes-free pose input technique using proximity sensing. The thumb-ring is built with a flexible printed circuit board hosting nine proximity sensors, which measure distances from the thumb to various parts of the hand. Because these distances vary depending on the user's hand position, ThumbTrak can distinguish which finger phalanges of the 12 phalanges the thumb touches in real-time using a support-vector-machine (SVM) model based on measured distances. The layout of the 12 finger poses (three phalanges per finger × four fingers) is similar to a T9 keyboard, which can be naturally extended to input numbers. A user study with ten participants showed that ThumbTrak could recognize these 12 micro finger poses with an average accuracy of 93.6%.

The contributions of our paper are:

- A novel thumb-ring-based input method which recognizes 12 micro-finger poses in real-time using proximity sensing and machine learning.
- A user study with 10 participants which evaluated the performance of the system (over 93%) and presented analysis on sensor numbers and positions.
- A discussion of opportunities and challenges about how to apply this technology in a wider range of real-world applications.

2 RELATED WORK

Gesture recognition has always been a classic research topic in the HCI community. In this section, we introduce and discuss prior work in wearable gesture input techniques involving one-handed input methods and ring-based approaches.

2.1 One-handed Gesture Input

Naturally, one-handed input allows users to input information with just one hand - often a more convenient option compared to interactions requiring both hands. A variety of one-handed input

technologies have been developed. Since armbands and wristbands are commonly worn accessories in society, many prior works use these two form factors to detect discrete hand or finger gestures. A variety of different sensing modalities have been employed to detect these gestures, including electric impedance tomography [36, 37], pressure sensing [5], acoustic sensing [6, 12, 18, 20, 35], Electromyography (EMG) [24, 28], magnetic sensing [14], motion sensing [24, 31, 32], electrical sensing [38], cameras [13, 23, 30] and proximity sensing [8, 9]. For example, Lu et al. [24] leveraged four surface electromyography sensors to recognize hand gestures such as grasping or opening the hand. However, prior research either required heavy instrumentation or mostly focused on recognizing mid- or large-scale one-hand gestures (e.g., hand movements, hand grasping), which lacks explorations on micro-finger one-handed poses (e.g., distinguishing which finger phalanges of the 12 phalanges the thumb touches).

2.2 Gesture Input with a Ring

Rings are another accessory that people are accustomed to wearing. Since they sit on the fingers (or thumb) themselves, they are arguably a more convenient method for sensing micro finger poses compared to wrist- or arm-related form factors. Consequently, many previous ring-based input techniques have been invented, recognizing discrete finger or hand gestures using a wide variety of sensing modalities. Previously explored modalities have included visual sensing (cameras) [3], motion sensing [11, 33, 34], acoustic sensing [35], magnetic sensing [1, 4], and infrared reflection sensing [26]. Ring-based input projects have also been invented to track continuous finger input on 2D surfaces [16] and thumb input in 3D space [4].

The most similar previous work to ours is FingerPing [35], which also recognizes thumb touches on the 12 finger phalanges. The system recognizes these poses via active acoustic sensing with a thumb-ring, and a wristband [35]. Compared to FingerPing, ThumbTrak only requires one piece of form factor (the thumb ring), which is arguably more practical, since it requires less instrumentation. Furthermore, ThumbTrak is potentially more comfortable comparing to FingerPing [35], given there is no vibration on the thumb. More importantly, FingerPing has a low cross-section performance, due to inconsistencies in ring and wristband placement, which results
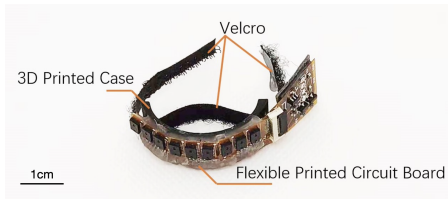
Figure 3: ThumbTrak Hardware

in inconsistent signals causing acoustic impedance to vary. Alternatively, ThumbTrak uses proximity sensing - a relatively more robust cross-session solution. We present a promising cross-session recognition performance report later on in the paper.

3 DESIGN AND IMPLEMENTATION OF THUMBTRAK

3.1 Pose Design

Inspired by the natural similarity of the layouts between the finger phalanges and the T9 keyboard [18], we designed 12 micro-finger poses (as shown by Figure 2), involving thumb touching on each of the three phalanges of the index, middle, ring, and little fingers. These micro-gestures have fewer movement distinctions comparing with other hand or finger gestures (e.g., fist and thumbs-up) and they allow input in a more discreet fashion for privacy. Furthermore, adopting an existing input layout is more straightforward to map the input poses to different functionalities in applications. For instance, the 12 poses construct a three (phalanges) by four (fingers) matrix, which can be used to input numbers. A subset of these 12 micro-finger poses can also be used as a directional pad.

3.2 Hardware Design

Figure 3 illustrates ThumbTrak's ring, which comprises three major components: 1) a flexible printed circuit board (FPCB), 2) a 3D printed case to provide support, and 3) a Velcro band. The FPCB board hosts nine proximity sensors (VL6180X)[1], and they operate at around 10Hz. We choose FPCB over a traditional printed circuit board (PCB) because FPCB is bendable and flexible, allowing it to wrap around the thumb to better measure distances in multiple directions. The more directions we can measure distances from, the more informative the sensor readings will provide. The FPCB is glued on a 3D printed case for more robust support. We use a Velcro band to fit the ring tightly and comfortably around thumbs of different sizes. The sensor readings are communicated from FPCB to a Teensy 3.2 using wired I^2C communication, which eventually sends the data to a python program running on a MacBook Pro for online classification via serial communication. We sequentially activate and change each proximity sensor's address to enable I^2C communication. In the device testing, we found that ThumbTrak consumes around 120mW power.

3.3 Data Processing Pipeline

We developed a Python program to continuously capture and analyze the stream of sensor data for real-time pose recognition. From
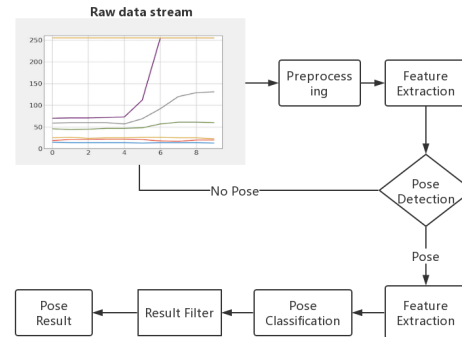
---

[1]https://www.st.com/resource/en/datasheet/vl6180x.pdf

Figure 4: Data Processing Pipeline

our internal testings between researchers, we found that the distance between the thumb and any given part of the palm did not exceed 150mm when performing candidate poses. If a distance value does exceed 150mm, we set its value to 150mm to prevent erroneous sensing results due to the out of range sensing distances, which signifies the sensor is out of range. Many out of range sensors are an indicator of "no-pose" status.

Every sample iteration, we gather each sensor's distance value for a total of nine initial features. We then use these nine values to derive two extra features: the number (quantity) of out of range sensors and the average value of all in-range sensors. This makes for a total of 11 features. We then feed all 11 features into a support vector machine model (SVM classifier with a linear kernel) to predict whether or not a pose is being performed. If the SVM model, which is responsible for segmentation, predicts a pose is being performed, we pass that segment values on to a second SVM model (with linear kernel) for classification. This second SVM classifier is responsible for predicting which finger pose of the 12 poses was performed.

To prevent potential system noises, we leveraged majority voting to detect the poses in real-time. For each data sample, we extract features, which are sent to the first SVM model to estimate whether it is a target pose or noise (non-pose). If the first SVM recognizes a data sample as a gesture, we activate the second SVM and classify the pose as one of the 12 micro-finger poses. At a specific time, if the previous 15 data points (1.5 seconds) are classified as poses, we then perform the majority voting. As the thumb may not be stabilized in the first half second, we only consider the classification result from the later one second for majority voting. Then we output the labels that represent the majority of the later 10 data points of the entire 15 data points as the gesture recognition result. Furthermore, if the SVM model outputs five consecutive results as noise (non-pose), ThumbTrak recognizes this as a reset for micro-finger pose recognition.

3.4 Calibration

Remounting the ring can introduce displacements on the sensor positions, which can impact the performance of the system. To address this issue, we introduce a calibration method in our system to help the users to remount the ring to its original position with minimal effort. We determined proximity to its original position by

outputting the absolute value of the difference between its current distance and its original distance for two poses: little-proximal and little-distal (the furthest fingers from the thumb). In the calibration process, we first asked participants to perform the two poses (i.e., little-proximal and little-distal). The researcher then empirically and iteratively adjust the ring position based on the output value. In our user evaluation, we set the average offset threshold value per sensor as 0.7mm.

4 USER STUDY

To evaluate ThumbTrak, we conducted a user study with ten participants with an average age of 22 (three female). All participants were right-handed. The user study comprises three main phases: 1) practicing phase—participants get to know the micro-finger poses and our prototype; 2) training phase—researcher collects training data with participants to generate our recognition model; 3) testing phase—participants are asked to perform different micro-finger poses to evaluate the performance of our system. Before conducting the user study with ten participants, we piloted our system and study process with two other participants. The study took around 60 minutes on average. The whole study procedure was approved by the institutional review board (IRB).

4.1 Practicing Phase

At the beginning of the user study, a researcher introduced our project and outlined the study's procedure. The researcher then helped the participant to fasten the ring, ensuring it be fixed tightly yet comfortably around the user's thumb to avoid unwanted rotation. Next, the researcher demonstrated how to perform each pose, instructing them to follow along with their dominant hand. After this, the participant familiarizes herself with different micro-finger poses. Participants were seated for the entirety of the experiment.

4.2 Training Phase

To collect training data, participants were asked to participate in three training sessions. In each training session, the participant was asked to perform five instances of each of the 13 poses (including "no-pose"). Poses were prompted by a screen displaying an image of an open hand with a red dot resting over the region of the finger to be touched. Images appeared for a total of four seconds: three seconds for participants to react and perform the pose and one second for the system to record the sensor data. All in all, each participant performed 65 randomly ordered poses continuously for each of the three training sessions. Between each training session, participants got a short break.

4.3 Testing Phase

Each testing session consisted of 60 poses (i.e., 12 micro-finger poses x 5). Additionally, after each completed pose, the real-time result was presented on the screen. If the result matched the requested pose, the interface turned green. Otherwise, the interface turned blue and displayed the falsely predicted pose. Every time when participants received the results, they were asked to reset the micro-finger pose to no-pose for segmentation purposes. The testing phase contains three sessions: we called the first two sessions as in-session tests, which focus on the overall performance of the system, and

the final one as the cross-session test, which mainly tests for the effects of remounting of the device. Between each testing session, participants got a short break. After finishing the last in-session test, a researcher helped the user to remove the ring. After a short rest, the researcher affixed the ring back onto the user's thumb and proceeded to undergo the calibration process. After calibration, the participant continued with the cross-session study, which was identical to in-session tests.

After completing all sessions, each participant was asked to fill out a questionnaire collecting some personal information regarding age, gender, hand dominance, wearable-device use, and thumb size. Participants also provided subjective feedback on the device, the comfort level, and perceived social appropriateness. Lastly, each participant received $10 for participating in the study.

5 RESULTS

Throughout the entire user study, all participants correctly performed every requested hand pose as the stimuli indicated on the screen, both in training and testing sessions. In total, across all 10 participants, we collected 1950 (13 poses x 150) instances in training sessions and 1800 (12 poses x 150) instances in testing sessions. Of the 1800 total testing instances, 1200 were in-session, and 600 were cross-session. If the recognition results in real-time matched the stimuli poses on the screen, we recorded it as a correct instance. Otherwise, it was a classification. The results reported in this section were the real-time performance recorded during the user study.

5.1 In-session Testing

Among the performance of ten participants, the average accuracy across all poses was 93.6%. The confusion matrix in Figure 5 shows the average accuracy results of all participants for each hand pose. Recognition was most accurate for poses involving the little finger and least accurate at the index proximal position (Figure 5). There

	index distal	middle distal	ring distal	little distal	index middle	middle middle	ring middle	little middle	index proximal	middle proximal	ring proximal	little proximal
index distal	96.74%	2.17%	0.00%	0.00%	1.09%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
middle distal	5.00%	95.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
ring distal	0.00%	1.00%	90.00%	9.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
little distal	0.00%	0.00%	0.00%	97.00%	0.00%	0.00%	0.00%	1.00%	0.00%	0.00%	0.00%	2.00%
index middle	2.06%	0.00%	0.00%	0.00%	96.91%	0.00%	0.00%	0.00%	1.03%	0.00%	0.00%	0.00%
middle middle	0.00%	0.00%	0.00%	0.00%	0.00%	96.00%	2.00%	0.00%	0.00%	2.00%	0.00%	0.00%
ring middle	0.00%	1.00%	1.00%	0.00%	0.00%	2.00%	89.00%	6.00%	0.00%	0.00%	0.00%	1.00%
little middle	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	99.00%	0.00%	0.00%	0.00%	1.00%
index proximal	1.02%	0.00%	0.00%	0.00%	4.08%	5.10%	0.00%	0.00%	85.72%	4.08%	0.00%	0.00%
middle proximal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.00%	0.00%	5.00%	90.00%	4.00%	0.00%
ring proximal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.00%	0.00%	0.00%	7.00%	89.00%	3.00%
little proximal	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.00%	0.00%	0.00%	0.00%	99.00%

Figure 5: Confusion Matrix for recognizing poses for in-session testing
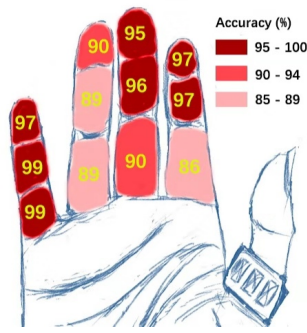
Figure 6: Accuracy for 12 Phalanges (Rounded to nearest integer)

is a common confusion among the proximal positions of the index, middle, and ring fingers. Specifically, we uncovered that index proximal (Accuracy=85.72%) was misclassified as index middle, middle middle and middle proximal (Figure 5). From analyzing these poses and feedback from participants, we found that some participants have a hard time performing the pose to touch the index proximal with thumb, because index proximal is the closest position to the thumb. Furthermore, we revealed that the "too close" situation made sensors have a smaller variability in distance comparing with fingertips, which results in the misclassification to index middle, middle middle, and middle proximal (Figure 5).

We then visualized the recognition performance on different finger positions (Figure 6), which depicts the accuracy for each of the 12 phalanges across all participants. Beyond the misclassification of the index proximal, we realized that the recognition performance of the ring finger is relatively worse than the middle finger and the little finger. In our study, we found that it is a natural behavior for participants to move their middle finger when they try to use their thumb to touch the ring finger, which may affect the recognition performance. In Figure 7, we present the overall performance for each participant during the in-session tests and the cross-session test. The figure illustrates that universally, all participants were able to achieve a relatively high average accuracy (Figure 7). Overall, we found that the highest accuracy achieved was 97.50% while the lowest was 88.79%. The median accuracy was 94.10%.
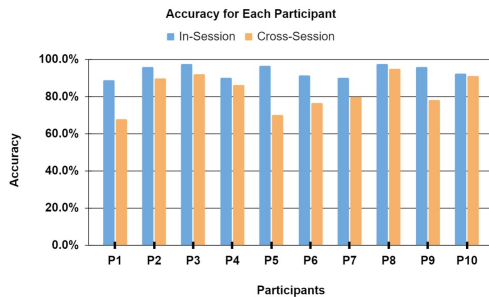

Figure 7: Accuracy for each participant

5.2 Cross-session Testing

In the cross-session testing, we found that the average accuracy across all participants was 82.7%. Cross-session confusion error distributions were quite similar to in-session errors. For example, we found that index proximal was also misclassified as index middle, middle middle, and middle proximal. Furthermore, we found middle middle was the second most erroneous pose and got misclassified as index distal, index middle, ring middle, and index proximal. Figure 7 shows that accuracy varies greatly among participants, from the highest 95.00% to the lowest 67.86%, with an 82.9% median accuracy. From the observation in the cross-session test when the researcher tried to assist participants in remounting the device, we discovered that the existing calibration process reduced the effort of memorizing the ring position relative to hand or finger biometrics (e.g., 'the ring was worn at 1cm below the thumbnail').
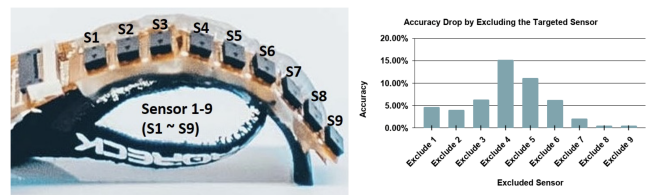

Figure 8: Left: Sensor name mapping on ThumbTrak. Right: Accuracy drop by excluding the targeted sensor

5.3 Analysis of Sensor Numbers and Positions

In our study, we utilized nine proximity sensors on the ring to recognize micro-finger poses. We then perform further analysis on how different positions and numbers of proximity sensors may affect the recognition performance. To understand the usefulness of different sensors at different positions, we first applied the study segmentation results and removed one sensor, and used the other eight sensors for training and testing, each time we change the position of the excluded sensor. From the analysis results, we uncovered that removing sensors in the middle caused the performance to drop the most. For example, the accuracy dropped 15.25% if we remove Sensor 4 (Figure 8). On the contrary, the performance stayed around 90% if we remove the sensor at the side of the ring.

To further understand the effect of using fewer sensors on the performance change, we leverage the study segmentation results and start with sensors in the middle, because we found sensors in the middle contribute to the performance the most. In this analysis, we start with Sensor 5; we then keep adding sensors surrounding the sensor group. Overall, We ended with five sensor subsets: Sensor 5, Sensor 4-6, Sensor 3-7, Sensor 2-8, and Sensor 1-9 (Figure 9). By analyzing associated performance changes (Figure 9), the accuracy stays around 40% when we only use Sensor 5 for training and testing (Figure 9). After we evaluate the recognition performance with using Sensor 3-7, the accuracy increases significantly to 85% (Figure 9). However, the accuracy increases less than 10% after we add the four sensors at the side (Figure 9). We further inspect the sensor output from the proximity sensors at the side, and we find that they sometimes remain out-of-range for certain micro-finger poses. Therefore, future research could perform a similar analysis to
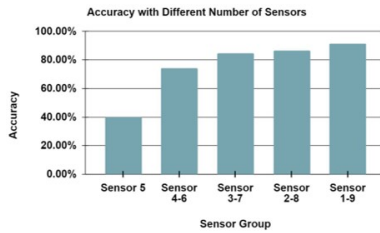
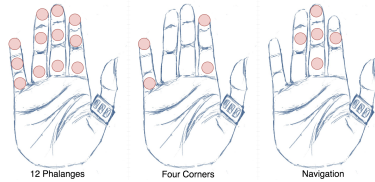**Figure 9: Accuracy with different number of sensors**

Figure 10: Layout of pose subsets

reach a compromise between power consumption and recognition accuracy.

5.4 Reduce Training session

In our in-session tests, we used data from all three training sessions to train the classifiers. However, it is often advantageous for users to reduce training data collection time. To investigate this, we reduced the training data to n training session(s) ($n = 1, 2$). This resulted in an accuracy of 83.5% for $n = 1$ and 91.2% for $n = 2$ (93.6% for $n = 3$). Thus, future developers should compromise between the effort of collecting training data and the recognition accuracy.

6 DISCUSSION

In the result section, we showed the accuracy and recognition confusion in both in-session testing and cross-session testing. We then analyzed the recognition performance effects of sensor numbers and positions. We also mentioned the usefulness of calibration when re-mounting the device and the tradeoff of reducing training sessions. In the discussion, we will further discuss 1) applications and opportunities for ThumbTrak, 2) potential affordance of ThumbTrak, 3) pose subsets for cross-session, 4) calibration and remounting process, and 5) hardware design improvements.

6.1 Applications and Opportunities for ThumbTrak

6.1.1 T9 Keyboard. ThumbTrak offers up to 12 discrete input gestures with an average accuracy rate of 93.6%. We found that it is natural to map these 12 micro-finger poses to a number input system due to their structural similarity (Figure 2). People could use ThumbTrak to make phone calls to their friends, and they could also utilize ThumbTrak as a T9 keyboard to accomplish text entry tasks in different messaging and social media applications.

6.1.2 D-Pad Control. Certain micro-finger poses can also be chosen to form new pose subsets (Figure 10) to meet the demand of different applications. Many applications do not require all 12 poses

to be functional. For instance, we can use a pose set (Figure 10) to control a D-Pad, which enables navigation through hierarchical menus for wearable devices. The buttons of the D-Pad can be naturally mapped to the index-middle, middle-distal, middle-proximal, and ring-middle positions (Figure 10). If we only include these poses in the training and testing sessions, this subset can achieve an average accuracy of 99.0% in post-analysis (Figure 11).

6.1.3 Music Control. ThumbTrak can detect the "pose" and the "no-pose" gesture. Thus, we can calculate the gesture holding time, which can provide continuous input, such as increase or decrease the music volume. The user could hold a pose to raise or lower volume and release the gesture when the volume has reached the desired level. Similarly, for music control, the user could use Thumb-Trak to place commands for 'Fast Forward', 'Rewind', 'Play', and 'Pause'.

6.1.4 Application Shortcuts. Another interesting pose subset, the four corner phalanges (index-distal, index-proximal, little-distal, little-proximal), can also be integrated as shortcuts with many applications. For example, the user could use the index-distal pose to do mobile screen capturing when playing a game on a smartphone (e.g., PUBG MOBILE). This subset presented a 97.1% accuracy in post-analysis (Figure 11).

6.1.5 Virtual Reality & Augmented Reality. Much existing virtual reality and augmented reality systems utilize WIMP (i.e., "Windows, Icons, Menus, Pointer") for user interactions. Similar to prior work on enabling faster control with finger or hand gestures [2, 19], ThumbTrak could also provide the opportunity of enabling fast hierarchical controls and reduce the effort of controlling the pointer movement through button and icon interactions, but without having to wear complex sensor instrumentation (e.g., sensor gloves [2]) or instrumenting a camera in front of the user [7, 19].

6.2 Potential Affordance of ThumbTrak

In our study, we only evaluated ThumbTrak on recognizing discrete micro-finger poses. However, in our experiment, we observed that the distance profile between thumb to other fingers could also be very informative to predict the thumb position in 3D space. One natural next step of ThumbTrak is to explore continuous tracking of the thumb position in 3D space using time-series modeling techniques, similar to the work of UTrack [4]. This would grant ThumbTrak an even richer expressiveness in input space. Furthermore, ThumbTrak could potentially support other hand or finger gestures, such as fist and thumbs up. Because the distance between the thumb and the palm would change when performing these gestures. Therefore, future applications could explore customized finger or hand gestures to naturally and unobtrusively interact with their interfaces with ThumbTrak.

In this current research, we focused on the general proof-of-concept and simply aligned nine sensors in a line. Beyond the existing hardware setup with a 1D array of proximity sensors, an extension of ThumbTrak could change the sensor layout from 1D to 2D to enable more depth detection. The angle of each sensor, their relative position to one another, and the number of sensors all may play a critical role in improving the ring's performance. Other than proximity sensing, future research could explore how
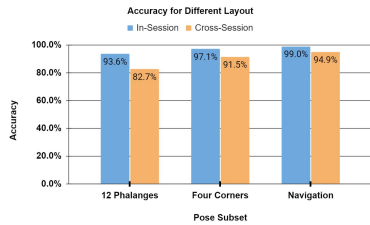
Figure 11: Accuracy for different layout of pose subsets

to leverage the micro depth camera on a ring-based sensing unit for hand/finger gesture recognition. Moreover, using radar sensors or acoustic sensors could also extend the potential applications of ring-based sensing units.

6.3 Pose Subsets for Cross-session

As illustrated above, some re-combined pose subsets are sufficient for multiple tasks and applications. This also applies to cross-section subsets. In our cross-session testing with 12 poses, the accuracy dropped from 93.6% to 82.7%. For navigation micro-finger pose subset (index-middle, middle-distal, middle-proximal, and ring-middle), we found that the accuracy only dropped 99.0% to 94.9% (Figure 11). In terms of the four corner phalanges micro-finger pose subset (index-distal, index-proximal, little-distal, little-proximal), the accuracy dropped from 97.1% to 91.5% (Figure 11). Thus, we found that using a subset of poses could enable the system to perform at high accuracy even if the participant need to remount the device, without applying any further calibration procedure.

6.4 Calibration and Remounting Process

In our calibration process, our algorithm design relies on the ring's position to remain stable. Once the ring shifts positions, the system may begin to misclassify hand poses. In our cross-session results, we found that the performance dropped around 10% after remounting the device. To better assist the calibration procedure, we may include more training data from different ring positions during calibration. This may compensate for the minor displacement of the ring and provide a more adaptive system to remounting situations.

Furthermore, we found that providing calibration feedback or adjustment recommendations is important for users while remounting their devices. To provide calibration feedback, we realized that it might be useful to attach small vibration motors to different positions of the ring to provide haptic feedback to users. Furthermore, the calibration process we used in the study—just by reporting the absolute differences of the sensor values comparing with the original position—may not provide enough information to a user where she should adjust the ring position. Therefore, having a visualization tool would be beneficial to users while they are using ThumbTrak in real-life scenarios.

6.5 Hardware Design Improvement

In order to ensure that all participants could wear our ring, we could not simply print one ring of a standard size. So we decided to create an adjustable ring. We achieved this by 3D printing a curved frame with a hole in each end. We then fed Velcro straps through both

holes so that when the strap is tightened, the curved 3D printed frame will flex and tighten around any sized thumb (as Figure 3 shows). An issue with this design, however, is its tendency to shift positions while users perform hand poses, reducing ThumbTrak's recognition accuracy as a consequence. In addition, this hard plastic PLA may not be comfortable enough for everyday uses. A potential solution could be to replace the plastic with other more elastic materials like rubber. Increased elasticity would result in a softer, more adjustable ring. Moreover, rubber would increase friction between the ring and the skin, helping the ring to remain in a fixed position. Also, it is worth noting that although it was important to make a one-size-fits-all model for the purposes of our experiment, rings by nature are not typically designed to be one-size-fits-all. In future form factors, ThumbTrak could therefore be integrated into customizable ring sizes in the same way that rings are custom-fitted in the jewelry industry today.

7 LIMITATION AND FUTURE WORK

In our paper, we conducted a quantitative study to asked participants to evaluate ThumbTrak in relatively static manners under a lab setting. However, people may leverage micro-finger poses while standing, walking, jogging, holding objects in their hands, or in complex environments, all of which were not included in the current study. Moreover, our current prototype requires wire connections to collect data for gesture recognition. In future work, we plan to first create a self-contained prototype that is self-powered and has wireless communications. We will further conduct scenario-based studies (e.g., [15, 21]) to both quantitatively and qualitatively evaluate the performance of ThumbTrack and identify potential problems under different contexts.

Another limitation is that we only included the "no-pose" gesture as participants' hands stay still in static situations. In real-life scenarios, people may accidentally move their fingers, which may result in different sensor outputs and unintended gestures. In our evaluation, we only studied discrete gestures under a lab setting, people might use micro-finger gestures in a continuous manner. Future work should explore methods to reduce false positives(e.g., activation gesture) and enable continuous gestures for real-life scenarios.

Furthermore, our researcher helped participants to calibrate the ring position in the cross-session testing, which limit our explorations on the behaviors of how participants remount the ring by themselves, which they will eventually do under real-life scenarios. In future work, we will conduct studies to specifically understand users' behaviors of calibration and remounting the device. Finally, our participants are mostly younger adults (average age of 22). In future work, we will include participants from different age groups. Although we understand our work could be strengthened with the above modification, we think our work—a novel thumb-ring-based input method that recognizes 12 micro-finger poses in real-time using proximity sensing and machine learning—makes novel contributions to HCI and mobile-sensing research community.

8 CONCLUSION

In this paper, we present ThumbTrak, a novel Thumb-ring based input technique able to recognize micro-finger poses in real-time

with proximity sensing. A user study with 10 participants shows that ThumbTrak can recognize thumb touches on all 12 finger phalanges with an average accuracy of 93.6%. We compared our in-session testing, and cross-session testing results and uncovered design guidelines of micro-finger pose recognition system with proximity sensing. We further discussed the potential applications for ThumbTrak (e.g., D-Pad), the accuracy of pose subsets for cross-session testing, calibration and remounting process, continuous tracking of thumb, and potential improvements of hardware design. We believe our findings would shed light on future research with micro-finger poses sensing.

ACKNOWLEDGMENTS

REFERENCES

[1] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2043–2046.

[2] Doug A Bowman and Chadwick A Wingrave. 2001. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings IEEE Virtual Reality 2001*. IEEE, 149–156.

[3] 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*. ACM, 549–556.

[4] 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*. ACM, 237–244.

[5] 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*. ACM, 161–166.

[6] Travis Deyle, Szabolcs Palinko, Erika Shehan Poole, and Thad Starner. 2007. Hambone: A bio-acoustic gesture interface. In *2007 11th IEEE International Symposium on Wearable Computers*. IEEE, 3–10.

[7] Mingming Fan, Zhen Li, and Franklin Mingzhe Li. 2020. Eyelid Gestures on Mobile Devices for People with Motor Impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–8.

[8] Rui Fukui, Masahiko Watanabe, Tomoaki Gyota, Masamichi Shimosaka, and Tomomasa Sato. 2011. Hand shape classification with a wrist contour sensor: development of a prototype device. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 311–314.

[9] 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*. ACM, 861–872.

[10] 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*. 553–563.

[11] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 14.

[12] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 453–462.

[13] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.

[14] 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*. 1526–1537.

[15] Mohammad Kianpisheh, Franklin Mingzhe Li, and Khai N Truong. 2019. Face recognition assistant for people with visual impairments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–24.

[16] Wolf Kienzle and Ken Hinckley. 2014. LightRing: always-available 2D input on any surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 157–160.

[17] 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*. ACM, 167–176.

[18] 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*. 1–13.

[19] Arun Kulshreshth and Joseph J LaViola Jr. 2014. Exploring the usefulness of finger-based 3D gesture menu selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1093–1102.

[20] 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*. ACM, 321–333.

[21] Franklin Mingzhe Li, Di Laura Chen, Mingming Fan, and Khai N Truong. 2019. FMT: A Wearable Camera-Based Object Tracking Memory Aid for Older Adults. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–25.

[22] Mingzhe Li, Mingming Fan, and Khai N Truong. 2017. BrailleSketch: A gesture-based text input method for people with visual impairments. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. 12–21.

[23] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proc. MobileHCI*, Vol. 10.

[24] Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. 2014. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE transactions on human-machine systems* 44, 2 (2014), 293–299.

[25] Adiyan Mujibiya, Xiang Cao, Desney S Tan, Dan Morris, Shwetak N Patel, and Jun Rekimoto. 2013. The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. 189–198.

[26] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 131–136.

[27] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2799–2802.

[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*. ACM, 167–176.

[29] Wei Sun, Franklin Mingzhe Li, Benjamin Steeper, Songlin Xu, Feng Tian, and Cheng Zhang. 2021. TeethTap: Recognizing Discrete Teeth Gestures Using Motion and Acoustic Sensing on an Earpiece. In *26th International Conference on Intelligent User Interfaces*. 161–169.

[30] David Way and Joseph Paradiso. 2014. A usability user study concerning free-hand microgesture and wrist-worn sensors. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks*. IEEE, 138–142.

[31] 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*. ACM, 3847–3851.

[32] 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*. ACM, 9–14.

[33] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 120.

[34] Cheng Zhang, Xiaoxuan Wang, Anandghan Waghmare, Sumeet Jain, Thomas Ploetz, Omer T Inan, Thad E Starner, and Gregory D Abowd. 2017. FingOrbits: interaction with wearables using synchronized thumb movements. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 62–65.

[35] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, et al. 2018. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 437.

[36] 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*. ACM, 167–173.

[37] Yang Zhang, Robert Xiao, and Chris Harrison. 2016. Advancing hand gesture recognition with high resolution electrical impedance tomography. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 843–850.

[38] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1491–1503.