RadarHand: A Wrist-Worn Radar for On-Skin Touch-Based Proprioceptive Gestures

RYO HAJIKA, TAMIL SELVAN GUNASEKARAN, and CHLOE DOLMA SI YING HAIGH, The University of Auckland, New Zealand
YUN SUEN PAI, Keio University Graduate School of Media Design, Japan
EIJI HAYASHI and JAIME LIEN, Google Inc., USA
DANIELLE LOTTRIDGE and MARK BILLINGHURST, The University of Auckland, New Zealand

We introduce RadarHand, a wrist-worn wearable with millimetre wave radar that detects on-skin touch-based proprioceptive hand gestures. Radars are robust, private, small, penetrate materials, and require low computation costs. We first evaluated the proprioceptive and tactile perception nature of the back of the hand and found that tapping on the thumb is the least proprioceptive error of all the finger joints, followed by the index finger, middle finger, ring finger, and pinky finger in the eyes-free and high cognitive load situation. Next, we trained deep-learning models for gesture classification. We introduce two types of gestures based on the locations of the back of the hand: generic gestures and discrete gestures. Discrete gestures are gestures that start at specific locations and end at specific locations at the back of the hand, in contrast to generic gestures, which can start anywhere and end anywhere on the back of the hand. Out of 27 gesture group possibilities, we achieved 92% accuracy for a set of seven gestures and 93% accuracy for the set of eight discrete gestures. Finally, we evaluated RadarHand's performance in real-time under two interaction modes: Active interaction and Reactive interaction. Active interaction is where the user initiates input to achieve the desired output, and reactive interaction is where the device initiates interaction and requires the user to react. We obtained an accuracy of 87% and 74% for active generic and discrete gestures, respectively, as well as 91% and 81.7% for reactive generic and discrete gestures, respectively. We discuss the implications of RadarHand for gesture recognition and directions for future works.

CCS Concepts: • **Human-centered computing → Gestural input**; *Mobile devices;*

Additional Key Words and Phrases: FMCW Radar, on-skin gesture, smartwatch gesture input, hand topography, deep learning, radar sensing, proprioception

ACM Reference format:
Ryo Hajika, Tamil Selvan Gunasekaran, Chloe Dolma Si Ying Haigh, Yun Suen Pai, Eiji Hayashi, Jaime Lien, Danielle Lottridge, and Mark Billinghurst. 2024. RadarHand: A Wrist-Worn Radar for On-Skin Touch-Based Proprioceptive Gestures. *ACM Trans. Comput.-Hum. Interact.* 31, 2, Article 17 (January 2024), 36 pages.
https://doi.org/10.1145/3617365

R. Hajika and T. S. Gunasekaran contributed equally to this work.

Authors' addresses: R. Hajika and T. S. Gunasekaran, The University of Auckland, 70 Symonds Street, Auckland, New Zealand; e-mails: ryo.hajika@auckland.ac.nz, themastergts007@gmail.com; C. Dolma Si Ying Haigh, D. Lottridge, and M. Billinghurst, The University of Auckland, Auckland, New Zealand; e-mails: chai915@aucklanduni.ac.nz, {d.lottridge, mark. billinghurst}@auckland.ac.nz; Y. Suen Pai, Keio University Graduate School of Media Design, 4-1-1 Hiyoshi, Kohoku-ku, Yokohama, Japan; e-mail: pai@kmd.keio.ac.jp; E. Hayashi and J. Lien, Google Inc., 1600 Amphitheatre Parkway, Mountain View, USA; e-mails: {eijihayashi, jaimelien}@google.com.

1073-0516/2024/01-ART17 $15.00
https://doi.org/10.1145/3617365

1 INTRODUCTION

The current gold standard for smart devices is touch, and a wide variety of touch-enabled wearable devices and wrist-worn devices are becoming more ubiquitous, such as smartwatches,[1] smart jackets[2] and even smart rings.[3] These wearable and wrist-worn devices allow instantaneous command input due to their location on the wrist. Through continuous contact with the skin, embedded sensors (like IMUs, microphones, and pulse sensors) record users' daily activities and monitor their health [57]. However, touch screens and touchpads in wearable devices lack any form of haptic feedback of distinct textures and surfaces, making them challenging to operate eyes-free.

The growing requirements in more ubiquitous scenarios continuously motivate researchers to design easy-to-use hand gestures and develop advanced sensing technologies [67]. Among the many options for command input, hand gestures, such as unimanual gestures [12, 40, 58, 75, 85], mid-air bimanual gestures [23, 46, 54, 61, 71], and on-skin gestures [9, 24, 75], require no tangible input devices and require minimal cognitive loads.

One method to address the limitation of distinct surfaces on a device is to allow a touch on the user's skin surface [7]. However, previous works on turning the skin into an interactive surface generally focused on the relatively flat areas of the body that are in proximity to hands, such as the palm or forearm [28, 35]. Alternatively, the back of the hand is comprised of distinct topographic features and surface texture, making it an ideal environment for interaction. The decision was made based on a human ability to perceive the locations of body parts without looking at them, referred to as *proprioception* [55]. Previous works showed that utilising skin surfaces that are rich in unique landmarks (such as joints, bumps, wrinkles, tattoos, and veins) can improve input performance with virtual items [8, 24, 35, 65]. One such topographically distinct region is the back of the hand.

Prior researches [28, 89] often require an additional sensor on the dominant or pointing hand or relies upon a camera-based system that is prone to occlusion, poor illumination, and privacy concerns. In other studies [13], researchers have used an on-the-shelf smartwatch to detect hand gestures through an integrated **IMU (Inertial Measurement Unit)** and a microphone. However, the microphone is often susceptible to background noise and privacy concerns, whereas IMU is vulnerable to motion noise and does not have spatial sensing capability to detect various hand gestures for on-skin interaction.

Previous research has not extensively explored the notion of sensing gestures at the back of the hand using the conventional gesture paradigm for smartwatches, including gestures such as tapping, horizontal swiping, vertical swiping, and rotation of knobs [16]. Earlier research that utilizes IMU shows promising results for tap gestures across the back of the hand [13]. However, it would be difficult to detect other gestures aside from tapping due to how IMU sensors use distinct vibrations to recognise gestures. In contrast, radar has shown impressive results in sensing a wide variety of gestures, has spatial resolution, and can achieve a small footprint to fit in a mobile device [4, 29, 38].

In this article, we present RadarHand, a wrist-worn miniature radar based on Google Soli that senses proprioceptive on-skin touch input gestures, which were designed upon our study on hand topography [38]. The topographic features of the back of the hand are distinct in terms of their position, length, texture, and angle. Taking advantage of them is suitable for realizing unique input modalities.

[1] https://www.apple.com/apple-watch-series-7/

[2] https://atap.google.com/jacquard/

[3] https://ouraring.com/

Additionally, Soli has multiple attractive properties for the deployment of gestural interactions in wearable computing. The **radar radio frequency (RF)** signal penetrates through plastic, glass, and other non-metallic materials, preserving its functionality inside device enclosures. It will also not be affected by ambient light or acoustic noise. Soli is also less privacy-invasive compared to image sensors since the radar does not produce distinguishable representations of a target's spatial structure. Finally, Soli's sensitivity to sub-millimetre displacements, regardless of distance, allows motion recognition in both near and far fields with the same hardware.

We have conducted three studies to evaluate our system. Prior to developing a prototype system, we evaluate the appropriate gesture design for RadarHand based on the previously established design considerations. To do this, we conducted an experiment to determine the performance of attaining landmarks on the back of the hand objectively. This reflects the proprioceptive and exteroceptive error in reaching the various landmarks on the back of the hand [53]. Our objective approach aimed to provide data to inform the design of the gesture set, as we aimed for a more rigorous approach than the current standard of using subjective scores (e.g., [9, 41]). We performed an evaluation to assess hand proprioception and tactile perception using an extended Fitts' Law predictive model [21]. We also ran a post-survey to determine user perception of gesture usability, memorability, social acceptance, and disambiguity. From this study, we obtained the least proprioceptive error landmarks using tapping gestures at the back of the hand that are suitable for gestural interaction in eyes-free and high cognitive load scenarios.

Leveraging the unique advantages of radar-based sensing and touch-based proprioceptive gesture insights from study 1, we developed a novel machine-learning model and collected a dataset to classify various gesture sets. First, we collected all the potential hand gestures data for our prototype, along with background gestures. Second, we developed and evaluated a novel machine-learning model based on the previous work [29] on radar gesture sensing. Finally, we introduce two types of gestures based on the locations of the back of the hand: generic gestures and discrete gestures [7]. Discrete gestures are refined to specific areas at the back of the hand, whereas generic gestures are able to be performed anywhere on the back of the hand. We grouped different sets of gestural groups based on generic and discrete gesture categories, which have the best accuracy and usage in various context-driven applications.

In our final study, we evaluated our trained machine learning model in real-time conditions and reported its false positives and shortcomings. We evaluated our RadarHand prototype and best-performing gesture groups from generic and discrete categories under two interaction modes: Active interaction and Reactive interaction. In Active interaction, the user initiates interaction with a system to achieve the desired output, such as tapping a button to access an app intentionally. In Reactive interaction, the system initiates interaction and requires the user to react, such as replying to a notification prompt.

Figure 1 depicts the overview of the three studies with the key information from each study steps.

Finally, we summarize all findings into design guidelines regarding the use of wrist-worn radars for gesture detection and interaction design.

This work introduces the following contributions:

(1) We introduce and demonstrate the use of radar as a compact wrist-worn device to detect on-skin touch-based proprioceptive gestures, which maintains privacy and is not affected by occlusion.
(2) We established the least proprioceptive error points on the back of the left hand for eyes-free and intuitive gesture interactions.

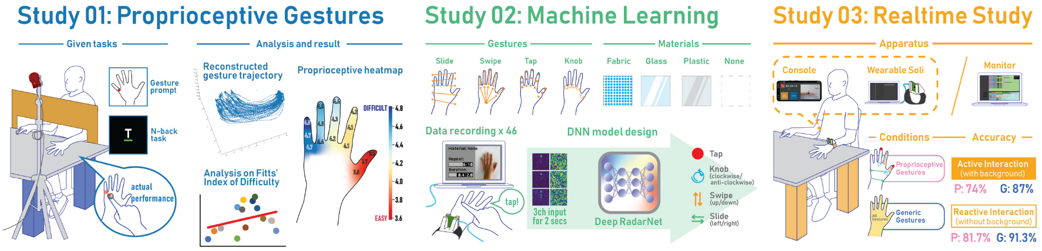


Fig. 1. RadarHand uses a wrist-worn radar to sense on-skin touch-based proprioceptive hand gestures. We investigated the proprioceptive and tactile perception nature of hand gestures at the back of the hand, gesture grouping from deep learning models, and the real-time performance of the prototype system. The teaser summarizes the methodology and results of three studies conducted in order to evaluate the RadarHand prototype.

(3) We group and classify the gestures using a deep learning model (accuracy of 92% for a generic 7 gesture set and 93% for the best discrete 8 gesture set) and propose contextual applications for each gesture group.
(4) We conducted a real-time evaluation on the model to report its performance (accuracy of 87% for a generic 7 gesture set, 74% for the discrete 8 gesture set in active interaction conditions, and accuracy of 91.3% for a generic 7 gesture set, 81.7% for the discrete 8 gesture set in reactive interaction conditions) and shortcomings.
(5) We propose a set of design guidelines regarding the use of wrist-worn radar for wearable gesture recognition, what can be applied, and what can be improved for further iterations.

2 RELATED WORK

2.1 Skin Surface as Input

A person's skin can act as an alternative input surface for wearable devices [78] due to its proximity to the device. Previous research has explored the skin as an input surface. For example, Skinput [28] is an armband that uses bioacoustic sensing to detect finger taps on both the forearm and hand. Minput [27] uses optical trackers placed under the wearable device allowing it to sense translational and rotational movements over different surfaces. Similarly, OmniTouch [26] mounts a depth camera with a pico projector on the user's shoulder, which projects onto the user's palm for interaction. SenSkin [50] uses infrared reflective sensors worn on the forearm to sense skin deformation and touch spots. Motivated by both of these works, Skin Buttons [35] combined laser projection with infrared sensors to turn the skin surface into virtual buttons. Finally, SkinWatch [49] further expanded on this by placing the infrared sensors under a smartwatch, turning the skin area near the watch into an input surface.

Associating sensors with wearable electric gadgets has become popular, as smartwatches are becoming increasingly mainstream while allowing for a myriad of new interactions for the forearm and hand. For example, SkinTrack [89] uses a ring that emits high-frequency AC (Alternative Current) signals alongside a sensing wristband for accurate finger touch coordinate tracking on the forearm and hand. LumiWatch [83] combines infrared sensors with a pico projector on a watch to enable continuous finger tracking on the forearm.

From this research, it can be seen that optical-based approaches are some of the most common solutions for detecting touch input on the skin. However, there are a few major drawbacks of optical methods; (1) there is a possibility of privacy concerns if the tracking uses computer vision technologies, and (2) occlusion can render it ineffectual.

Besides optical solutions, acoustic solutions have also been used. For example, SonarWatch [37] uses ultrasonic sensors mounted on a watch to detect interactions on the forearm. PUB (Point Upon Body) [39] also used ultrasonic sensors for forearm interactions. However, ultrasonic waves often get obstructed in environments with varying sound levels and temperatures. Lastly, other forms of sensing include tattoo-based wearables [34, 80] or stretchable materials [47, 77] attached to the skin. In spite of the sensing robustness, these solutions require the user to directly attach material to the skin itself, which can be uncomfortable and impedes the tactile sensation of the skin [48].

The closest related work is TapSkin [87] and Taprint [13]. Tapskin uses an IMU and a microphone to detect gestures at the back of the hand. However, such acoustic sensing also leads to potential privacy concerns. For Taprint, the IMU detects tap gestures at the knuckles of the hand. An IMU senses tiny vibrations on the skin surface as opposed to actually understanding the space, and proximity between gesture points may lead to misclassification. On the contrary, radar is primarily designed to detect proximity, allowing us to include a wider range of gesture sets on more interaction points.

2.2 Body Landmarks for Interactions

Some previous works have used distinct hand topography instead of flat surfaces as an interaction surface. Hand topography is body landmarks on the hand that are tactually or visually distinct from their surroundings [79]. The key benefits of targeting body landmarks is that it allows for more accurate localization of interactive elements, provides guidance through affordances and constraints, and allows for mapping of functionality [65]. Landmarks are even useful for placing and recalling virtual items due to their distinctiveness [8].

Previous research found that (1) observation of the hand while performing an action, (2) tactile cues sensed by the palm, and (3) tactile cues sensed by the pointing finger contribute to hand and palm-based interfaces [24]. This means that the tactile sensation on the palm and pointing finger could reduce proprioceptive error, especially when visual cues are absent. Gustafson et al. [23] used this finding to develop an imaginary phone interface where the right index finger acts as the pointer and the left palm acts as a smartphone display in a grid-style layout. Compared to the palm, though, the back of the hand contains more distinct skeletal landmarks like the knuckle bone as well as different textures like the nail surface, which is smoother. Depending on the state of the hand, the deformation of the skin and skeletal components are so distinct that they have proven to be useful for sensor detection [68, 81].

A more recent work, SkinMarks, defined interactions on the body using conformal skin electronics [79]. The interactions were performed on distinct topographical features such as skeletal landmarks like the knuckle bone and even on accessories like rings. However, similar to the tattoo-based wearables, these methods impede the tactual sensation since it covers the skin surface area. To overcome surfaces that lack distinct landmarks and are usually covered with clothes, FabriClick [20] looked into integrating push buttons into the fabric. This essentially increases the tactility of clothing-based interactions. However, our research looks into direct interactions with the skin surface and landmarks.

2.3 Assessing Proprioception

Humans typically have much higher visual dominance over other forms of sensory feedback [10]. Outside the typical five senses (sight, sound, touch, smell, and taste), the “sixth sense” is often known as proprioception [62]. Sometimes called kinaesthesia [5], it is defined as the sensation of body and movement that is typically absent from conscious perception [70]. Proprioception relies on the mechanosensory neurons which are often found in the muscles, tendons, and joints [18]. This allows us to still operate even without visual stimuli, with the most common substitute being the hands. The human hand, in particular, is not only dense with cutaneous mechanoreceptors. It is also capable of performing minute and micro motions like holding a pencil to write with [19, 32, 44, 69]. The forearm, followed by the hand and fingertip, has an increasing level of cutaneous mechanoreceptors [44, 48]. This partially inspired us to focus on our research on the proprioception of upper limbs and hands, in which we utilize one hand as an input motion/pointer and another as a surface for interaction.

Research to assess proprioception in the human body started as early as 1860 [17] with a study assessing the amount of force required by the limbs to overcome gravity while lifting weights. Wycherley et al. [82] developed a portable device used to measure joint position sense, specifically the metacarpophalangeal joint of both the index fingers. This device was also used by Smitt et al. [63] to measure proprioception for musicians and dancers, particularly string players. However, the device has several limitations, namely, it relies on custom hardware and only measures the index fingers for both hands. The proprioception of the ankle has also been studied, since it is critical for maintaining balance [33]. Han et al. [25] and Hillier et al. [30] provided an in-depth review on the various methods to assess proprioception. For RadarHand, we look into a recent method proposed by Gunasekaran et al. [21] that used Fitts' law and the N-back task to evaluate proprioception and tactile perception.

2.4 Radar Sensing

The first radar systems were developed in the 1930s. It has been commonly used for detecting large moving objects like aircraft and ships [76]. Since that time, radar sensors have shrunk in size. The recent development of miniature radar sensors like Soli [38] has enabled radar-based, precise motion sensing to be integrated into a broad range of everyday consumer devices. These sensors support micro-gestural interaction with a hardware in a small footprint, low power consumption, and less privacy concerns compared to other gesture sensing techniques. Wang et al. [73] introduce a range of possible micro gestures that can be detected with Soli. Attygalle et al. [4] also explored micro gestures using wrist-worn radar sensors when interacting with objects.

Radar can also penetrate through some materials and elicit insights of a target object. RadarCat [86] explored the potential of applying millimetre wave radar signals to detect and classify different objects placed on top of the sensor hardware. Radar functions as a non-invasive sensor that can classify the target object material from the profile of the reflected signal. There are also some uses of the radar in healthcare monitoring, medical imaging, and vital sign measurement [59]. Other uses of radar are for respiration detection [56], sleep pattern analysis [90], presence detection [51], tangible interaction [22] and human activity recognition [74]. Similarly, Ahuja et al. [3] converted video signals of human activity to radar Doppler signals for the purpose of recognizing human activity.

Some consumer devices have already adopted radar sensing, such as the Google Pixel 4 smartphone with Soli,[4] and the Nest Hub with presence detection.[5] However, the gestural interaction with these devices is currently quite limited since those systems focus on macro-gestures, such as omnidirectional swipes or a "tap" gesture in mid-air.

We explore wrist-worn radar for on-skin touch-based proprioceptive gestures in RadarHand toward smartwatch interaction. We have summarized the on-skin sensing wearable literature works in Table 1. To our knowledge, we are the first to propose radar-based gestural input for a standalone wrist-mounted wearable for smartwatch interaction, in conjunction with a novel gesture paradigm that uses hand topography for proprioceptive and haptic cues.

[4] https://store.google.com/us/product/pixel_4

[5] https://atap.google.com/soli/products/#nest-hub

Table 1. On-Skin Sensing Wearable Interface Literature Survey

Related Works	Sensing	Privacy	Reliable and Robust	Always - on	Small-footprint	Low- Computation Cost	Spatial Sensing
Skinput [28], Sonarwatch [37], Viband [36], PUB [39]	Sound/Acoustic	✓	X	X	✓	✓	X
Tapskin [87], IDial [88]	Microphone	X	X	X	✓	✓	X
Watchsense [64], Omnitouch [26] , Imaginary Phone [23]	Vision	X	X	X	✓	X	✓
LumiWatch [83], Senskin [50], Skinwatch [49], Skin Buttons [35]	Infrared	X	✓	✓	X	✓	X
Taprint [13], Tapskin [87], IDial [88]	IMU	✓	✓	✓	✓	✓	X
Hambone [15]	Piezoelectric	✓	✓	✓	✓	✓	X
wristflex [14]	Pressure sensing	✓	✓	✓	X	✓	X
Skintrack [89]	Electrical Wave-guide	✓	✓	✓	X	X	X
RadarHand	Radar	✓	✓	✓	✓	✓	✓

3 RADARHAND BENEFITS AND DESIGN CONSIDERATIONS

When considering gestural interaction in wearable computing contexts, we need to consider a different set of technological and interaction requirements. Hayashi et al. [29] pointed out that users will interact with such devices at the periphery or even out of their visual attention. These points clearly distinguish the benefits of the use of radar over other conventional sensing methods:

(1) *Always On.* Gesture recognition techniques should run continuously and be ready anytime whenever the user wants to initiate the interaction. The core value of gestural interactions in wearable computers is in their immediacy: the user can quickly interact on a small device for simple tasks with minimal cognitive effort and without complex hand-eye coordination. Any friction, such as the need to wake up the device, will impact the usefulness of gestural interactions in these contexts. We refer to this as active interactions, which we elaborate further in Study 3.

(2) *Reliable and Robust.* Gesture recognition techniques for wearable computing applications should work in various contexts. They should be robust against environmental changes such as sensor coverings, atmospheric temperature, and lighting conditions.

(3) *Private.* The diffusion of products with advanced sensing capabilities in personal spaces, such as our bedrooms, living rooms, or workplaces, makes privacy a key factor for their wide adoption.

(4) *Potentially Small and Invisible.* Gesture recognition techniques should possess a sufficiently small footprint to be embedded in various objects without compromising their form factor or aesthetic appeal. The interaction space should also be compact enough to accommodate gestures in close proximity or involve micro-movements. These techniques should be integrated seamlessly behind surfaces without necessitating openings or other modifications to the product's physical design. They should demonstrate the capability to penetrate various materials, including those commonly found in everyday objects, such as glass or cloth. This aspect is explored in detail in Study 2.

(5) *Low-Computation Cost.* The computation cost of gesture prediction should be efficient and lightweight to be executed on wearable devices. We explore an interaction mode called reactive interactions that is better for battery life which we elaborate further in Study 3.

To design the RadarHand gestures, which take advantage of the proprioceptive and tactile perception sense, we considered several factors:

(1) *Dominant Hand as a Pointer, Non-Dominant Hand as a Surface.* We aim to have RadarHand's interaction depend on a finger of the dominant hand (assigned as the right index finger in the prototype system) as a pointer and the surface of the non-dominant hand (assigned as the left hand) as an interactive surface. In general, the dominant hand is more accurate and intuitive when it comes to performing specific gestures [42].

(2) *Landmark-Driven Gestures.* As one of the primary goals of our gesture design is to be proprioceptive, we focus on the gestures being based on hand topography [65]. The section of the hand with the most distinct landmarks becomes the targeted surface of interaction.

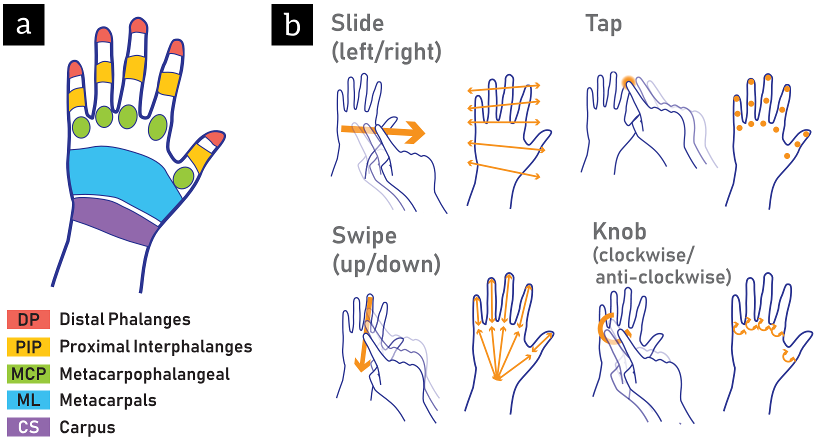


Fig. 2. (a) The gestural interaction area around the back of the hand, (b) RadarHand gestures

(3) *Open Hand State.* The state of the non-dominant hand is kept open with each of the fingers stretched for the gestures. This is to accommodate more of the aforementioned touch-based proprioceptive gestures, such as long slides along a finger, gestures on a nail, and so on.
(4) *Explicit Gestures.* The selected gestures are explicit, which means they need to be purposefully performed, as opposed to implicit gestures, which are potentially "normal" everyday actions such as waving, clapping, and so on [6]. This is to reduce overall false positives and negatives.
(5) *Gestures Derived from Smartwatches.* We refer to a gesture design when interacting with smartwatches.[6] Smartwatches typically have three basic interactions, which are tapping, vertical slide, and horizontal slide [16]. Additionally, there is also a rotational gesture which refers to rotary input in certain smartwatches, as mentioned by the Android Wear developer guide.[7]

From these considerations, we target the back of the non-dominant hand as the interactive surface due to its many distinct landmarks, including nails, fingers, knuckles, and so on. We support four distinct gestures, which are (1) tapping, (2) vertical slide, (3) horizontal slide, and (4) knob rotation [16, 42]. We focus on distinctive points like nails or **distal phalanges (DP)**, middle knuckle or **proximal interphalanges (PIP)**, knuckles or **metacarpophalangeals (MCP)**, the back of the palm or **metacarpals (ML)**, and wrist or **carpus (CS)** Figure 2(a). Gestures such as tapping ML or CS are removed since there is no distinct landmark to tap. Based on our gesture and the location area at the back of the hand, we established 55 gesture possibilities Figure 2(b). For tapping, 15 tapping gestures were selected, ranging from tapping on the pinky MCP to the thumb DP. For slide, we looked at sliding on each of the fingers in both directions, for a total of 10 gestures. Another 10 sliding gestures were selected on the ML for each finger. For knob rotation, we only perform this on the MCP of each finger for both clockwise and counter-clockwise, for a total of another 10 gestures. Lastly, a horizontal slide is performed along the CS, ML, MCP, PIP, and DP for both directions for a total of 10 gestures (15 + 10 + 10 + 10 + 10 = 55).

[6] https://developer.apple.com/design/human-interface-guidelines/inputs/touchscreen-gestures
[7] https://developer.android.com/training/wearables/user-input/rotary-input

4 STUDY 1: PROPRIOCEPTIVE GESTURES

Before proceeding with the development of the RadarHand prototype, it is crucial to evaluate the appropriate design of gestures based on established design considerations. In this context, identifying the landmarks on the back of the hand with the least proprioceptive error becomes essential in order to design effective gestures specifically for these landmarks. Proprioception plays a vital role in perceiving the position and movement of our body parts, including our fingers. Exteroceptors aid in verifying the accuracy of touch locations when tapping on the back of the hand. Therefore, we objectively assess the performance of reaching different landmarks on the back of the hand, as this directly reflects the associated proprioceptive error [53].

In order to adopt a more rigorous approach compared to the subjective scoring methods employed in previous related studies (e.g., [9] and [41]), we conducted experiments to objectively measure the performance of reaching landmarks on the back of the hand. This objective approach aims to provide empirical data that can inform the design of the gesture set. By employing this approach, we aim to obtain precise and reliable insights into identifying the landmarks on the back of the hand that exhibit the least proprioceptive error, making them most suitable for gestural interactions. These identified landmarks, with the least proprioceptive error, will serve as the optimal points on which users can perform gestures under conditions of high cognitive load and eyes-free scenarios. This will enable enhanced smartwatch interaction for everyday activities.

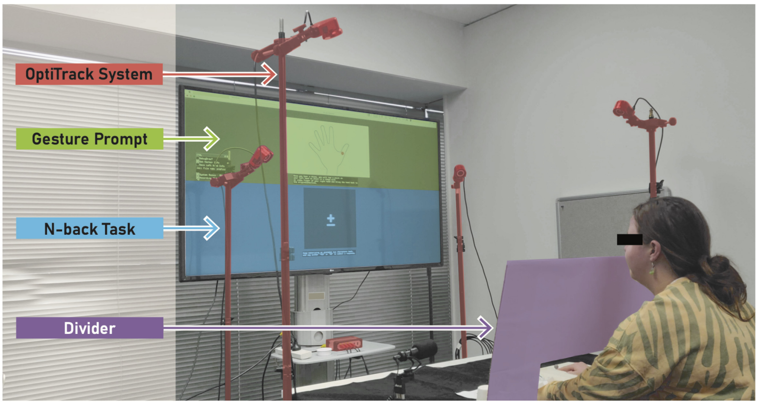
4.1 Apparatus

Figure 3 shows the study setup. Two computer systems were running simultaneously to conduct the session. The first system was for precise hand tracking. We set up a motion tracking system with four OptiTrack Flex 13[8] cameras to track retroreflectors that are applied to the index fingertip of the right hand. The motion capture cameras were placed around the main table and were connected to a Windows desktop computer, which ran OptiTrack Motive[9] software for real-time data collection. The second system was for an N-back task and gesture prompting. A 15-inch MacBook Pro laptop computer with a 65-inch generic colour monitor was used to display both the N-back task and gesture prompt with an illustration of the left hand annotated with a red dot. The red dot on the hand illustration tells the participant the exact position to perform a tap gesture. Since both systems were linked through a local network connection, live motion tracking data of the right index fingertip was streamed from the first system to the second in real-time. The second system ran C++-based custom software built with openFrameworks,[10] which prompted gestures with graphics, ran the N-back task, and recorded live motion data of the right index fingertip and the N-back task results.

We also introduced physical constraints to the setup to keep it the same across all the participants. To remove any chance of visual aid, we placed a large piece of cardboard attached to the edge of the table as a divider to obstruct the participants' vision of their own hands.

For reliable hand motion tracking, we used 3D-printed hand placeholders for the left hand. The placeholders were attached to a piece of paper with a hand outline printed on it, which was attached to the table. For the right hand, we sculpted a foam block that was attached to the table so that the participant could place their hand in it with the index finger pointed outward. These solutions also ensured that the hands were returned to the original position after each task without the need for looking. All apparatus was sanitized before and after each participant.

[8] https://optitrack.com/cameras/flex-13/
[9] https://optitrack.com/software/motive/
[10] https://openframeworks.cc/



[redacted]

Fig. 3. System design for the Study 1 setup, to understand the proprioceptive nature of the hand gestures: (1) the OptiTrack motion capture system observes the interaction area, (2) gesture prompt, and (3) N-back task visuals are displayed on a monitor, (4) a divider that prevents study participant from looking at their hands directly

4.2 Study Design

The design of our study is based on the methodology proposed by Gunasekaran et al. [21] that adapted Fitts' law and N-back task for assessment of proprioception. In the first stage of the study, we focused on finding the best locations that have the least proprioceptive error to perform hand topology-based gestures while balancing a high cognitive load. This was done by getting the participants to perform tapping gestures with their right index finger on different areas of the back of their left hand and then analysing the precision of the gesture performance on the target spot using Fitts' Law as the means of evaluation.

4.3 Participants

A total of 26 participants (13 females, mean age (years): 25, SD: 3.2) were recruited for this study. Almost a quarter (24%) of the participants used a laptop on a daily basis, and 27% used smartwatches. All the participants were right-hand dominant with no prior accidents that could affect motor skills in the hand.

4.4 Procedure

The participant first read and signed the consent form, followed by the information sheet about the N-back task while the experimenter explained the tasks. The width of each finger on their left hand was measured with a vernier calliper. Then, fifteen spherical retroreflective markers of 10-mm diameter were attached to the MCP, PIP, and DP of each finger on their left hand. One spherical marker was attached to the index DP of their right hand. The location of the markers was then captured using the OptiTrack system, where the system retrieved the three-dimensional position of the points of interest. The markers were removed from the left hand after the positions of the markers had been captured so that participants could actually touch the MCP, PIP, and DP points on their fingers.

Participants initially begin a short practice round to familiarize themselves with the system before moving on to the actual study. As shown in Figure 3, the N-back task was running on the bottom half of the main screen. For each section of the N-back task, an alphabetical character was presented for 500 milliseconds. Then, the participant had 4,500 milliseconds to answer verbally if that character was the same as the one or two characters shown before. On the top half of the screen, a gesture prompt with the left-hand silhouette and a red dot, shown somewhere on one of the MCPs, PIPs, and DPs on the left hand, was displayed. The participant would tap the corresponding position of the left band with their right index fingertip, then return the right hand back to the resting position indicated with the foam hand rest. They were asked to answer the N-back test and perform the tapping gesture simultaneously. They would tap each dot position three times during the study, and there were 45 tapping gestures performed overall (15 positions x 3 iterations). The entire gesture performance section took 5 minutes or less time. Each finger movement tracked with the OptiTrack system was recorded, along with the response and answering time of the N-back test.

Later, participants were required to complete a questionnaire that examined four different gestures (tapping, vertical slide, horizontal slide, and knob rotation) on the free hand state. Participants were shown short videos of the gestures and encouraged to try them out. The questions asked were around the usability, memorability, and social acceptance for each of the gestures [42]. Participants were also asked to rank their preferred finger and gestural design area. The whole experiment took around an hour to complete. At the end, participants were compensated with a $10 shopping voucher.

4.5 Computing the Fitts' Formula

The movement data consisted of the position coordinate, timestamp, velocity, and the marker's angular velocity. The N-back response data consisted of the participant's answer and the timestamp. We used the Fitts' law extension of Murata and Iwase [45] and Cha and Myung [11] to compute the **Fitts' Index of Difficulty (ID)**. To achieve this, we made a Python script to compute the distance parameter (A), width parameter (W), and movement time (MT). Based on the measured finger width (W), we created a bounding box around the target point, shown in Figure 4 on the right. At every sampled data frame, we measured the traversal distance between the right index and the center of the bounding box. From there, we could calculate the minimum distance (A) for when the right index entered the bounding box before returning to the origin. MT was the time taken during the right-index trajectory movement toward the target.

According to the revised Fitts' law of Murata and Iwase [45], the inclination angle (θ_1) is formulated using trigonometric calculations of the minimum distance point (t), origin point (o), and ground plane (n) as these points form a triangle, as shown in Figure 4. Using these three points, we calculated the inclination angle using the formula below:

$$\theta_1 = \arcsin \frac{o - n}{o - t} \tag{1}$$

For the revised Fitts' law of Cha and Myung [11], we used a trigonometric evaluation of three points to compute the azimuth angle (θ_2) value: the origin point (o), the ground plane (n), and the displacement of the minimum distance point of the original plane (t'). These points formed a triangle, shown in Figure 4 from which we calculated the azimuth angle using the formula below:

$$\theta_2 = \arcsin \frac{n - t'}{o - t'} \tag{2}$$

4.6 Results

4.6.1 Analysis of Fitts' ID vs MT. Figure 5(a) and Figure 5(b) shows the comparison between the two Fitts' IDs of Murata and Iwase [45] and Cha and Myung [11]. It can be seen that it has a stronger linear relationship compared to Murata and Iwase's Fitts' ID when Cha and Myung's ID linear regression value is closer to 1, though both did not perform very well with our data. The

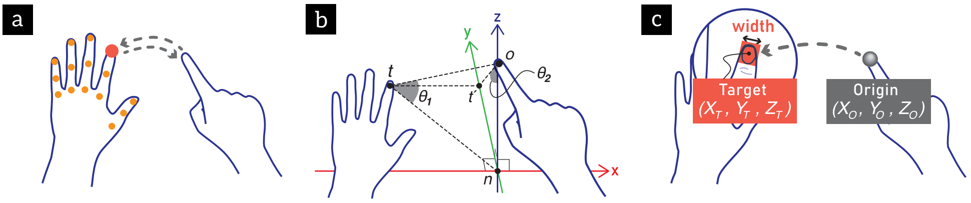


Fig. 4. (a) Locations of tapping gestures, (b) Azimuth angle and Inclination angle computation, (c) Target size and minimum distance point

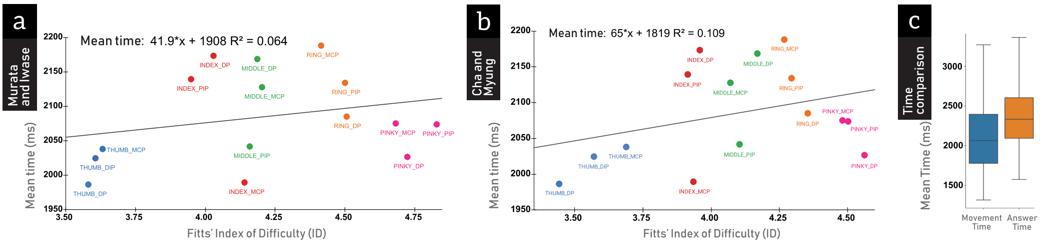


Fig. 5. The relationship between mean gesture performance time and ID (Fitts' Index of Difficulty) using the method proposed by Murata and Iwase (a), and that of Cha and Myung (b). We also compared MT (Movement Time) and the N-back task answering time (c).

Fitts' IDs were mainly affected by distance and target size conditions. The effect of distance shows MT gradually increases as the target fingers move from the thumb to the pinky. The variance in target size due to the different finger widths of the participants was also a reason for the non-linearity in Figure 5 between MT versus ID. The data was modeled using the revised Fitts' ID formula for the pointing tasks in 3D space [45]. As a result, we got ID levels for each of the 15 different conditions based on the equation. The model regression coefficients were $R^2 = 0.0604$ for Murata and Iwase's method and $R^2 = 0.109$ for Cha and Myung's method (computed based on the Fitts' formula detailed in Section 4.5). From the ID calculated using Murata and Iwase extended Fitts' law, we created a heat map that depicts the distribution of Fitts' ID from thumb to pinky, as shown in Figure 1 (right).

4.6.2 Analysis of Movement Time vs. Answer Time. Since the participants were required to perform the tapping and N-Back tasks simultaneously, there could be a delay in one task. This will allow us to understand user behavior as well as performance accuracy while performing a task under high cognitive load. **Movement time (MT)** is the time taken by participants to move from the origin to the target. **Answer Time (AT)** is the time taken by the participant to give an answer for the N-Back task. The rightmost plot in Figure 5 depicts the box plot comparing MT and AT with the y-axis represented in milliseconds. There is a general trend among all the participants that they try first to touch the target and then give an answer to the N-Back task. Running a paired t-Test gives $t(6.07) = 478, p < 0.001$, which shows there is significance between MT (M = 2128.4, SD = 452.4) and AT (M = 2360.5, SD = 380.8).

4.6.3 Questionnaire Results. For the post-experiment questionnaire results, the average scores for all the hand gestures are shown in Figure 6. We first tested our data for normality using the Shapiro-Wilk test and found its distribution is not being normal. As our data is non-parametric, we ran the Friedman test and found statistical significance across usability ($X^2(3) = 57.758, p < 0.001$),

Table 2. Ranking of Gestures based on User's Preference

Gestures	Finger	Hand Position
Tapping	Thumb > Index > Middle > Ring > Pinky	MCP > DP > PIP
Knob Rotation	Thumb > Index > Middle > Ring > Pinky	-NA-
Vertical Slide	Index > Thumb > Middle > Ring > Pinky	MCP-DP > CS-MCP
Horizontal Side	-NA-	CS > ML > MCP > PIP > DP

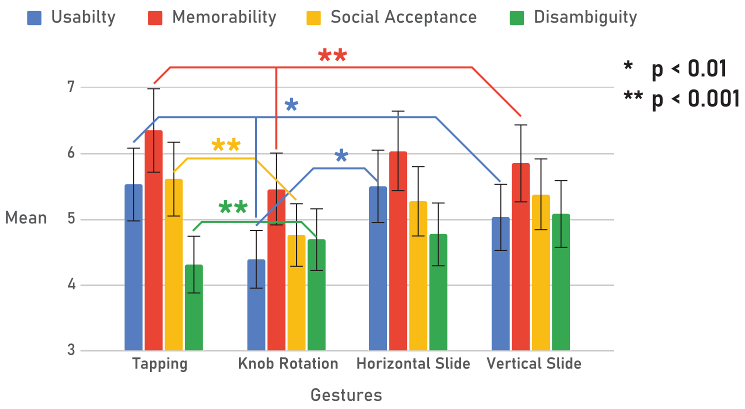


Fig. 6. The results from the survey by gesture category, grouped by the type of gesture performed.

memorability ($X^2(3) = 36.482, p < 0.001$), social acceptance ($X^2(3) = 34.119, p < 0.001$), and disambiguity ($X^2(3) = 28.702, p < 0.001$). Looking at usability first, we conducted a Nemenyi post-hoc test and found that the usability interaction on tapping gesture and knob rotation, tapping gesture, and vertical slide ($p < 0.01$) lead to significance. Horizontal slide and knob rotation ($p < 0.01$) also lead to significance under usability. For memorability, we found that the tapping gesture and knob rotation, tapping gesture, and vertical slide lead to significance ($p < 0.001$). Next, under social acceptance, the tapping gesture and knob rotation ($p < 0.001$) lead to significance. Finally, knob rotation and tapping gave significance under disambiguity ($p < 0.001$). All other combinations lead to no non-significant results ($p > 0.05$). The user preference ranking of fingers, hand position, and gestures are summarized in Table 2.

4.7 Discussion

Our findings indicate that both Fitts' models exhibit a non-linear relationship. When the index of difficulty (ID) is high, reaching the tap point becomes more challenging, resulting in decreased gesture accuracy. In terms of the fingers of the left hand, the ID gradually increases from the thumb to the pinky, which aligns with the sensory somatotopic mapping of the human hand. This observation may potentially explain the variation in the Fitts' ID between fingers. Notably, the index and thumb fingers have shown a larger activation volume in the somatosensory cortex during an fMRI study [43], with the index finger exhibiting the most prominent activation cluster.

Regarding gesture locations on the fingers, excluding the thumb, the ID is lowest at the metacarpophalangeal (MCP) joint, followed by the proximal interphalangeal (PIP) joint, and finally, the distal phalangeal (DP) joint. Although the fingertip has the highest number of neuron endings, it exhibits a higher ID compared to the other joints. This can be explained by two factors. First, according to Long et al. [41], individuals perceive their bodies, especially their fingers, to be smaller when there are no visual cues. Second, the palms have a higher concentration of muscle mass compared to the fingers, resulting in a greater number of mechanoreceptors.

Our study involved participants concurrently performing two tasks: touching the target point and answering the N-back task, without relying on visual cues. From the analysis of the average touch (AT) time for the N-back task and the movement time (MT), we observed a trend where participants prioritized touching the target before engaging with the N-back task. This observation can be attributed to the fact that proprioceptive-based tasks generally impose lower mental load compared to the N-back task [2]. Consequently, this suggests that users may still be able to accurately perform tapping gestures without visual aid using RadarHand in real-world scenarios where they can briefly focus on interacting with the wearable, such as when driving or carrying objects that obstruct their vision.

The survey results, depicted in Figure 6, and the analysis of the questionnaire revealed significant differences in perception among the different gestures. The tapping gesture was perceived as significantly more usable than knob rotation and vertical slide gestures. Furthermore, the horizontal slide gesture received significantly higher usability ratings compared to knob rotation. In terms of memorability, participants perceived the tapping gesture as more memorable overall when compared to the knob rotation and vertical slide gestures. Additionally, the tapping gesture received a significantly higher score in terms of social acceptance compared to the knob rotation gesture. Finally, participants indicated that the knob rotation gesture was significantly more ambiguous than the tapping gesture.

Analysis of our results, as depicted in Figure 5, aligns with the perceived finger and joint ranking. Consequently, based on the survey and Fitts' study, we have determined that the thumb, index, and middle fingers are the least prone to proprioceptive errors and are therefore preferred for gestural interactions. Specifically, the metacarpophalangeal (MCP) and distal phalangeal (DP) joints are identified as the least prone to proprioceptive errors and are preferred locations for performing gestures.

In our current empirical study design, participants were instructed to tap on the back of the hand under conditions of high cognitive load and without visual cues. This tap gesture process incorporates both proprioceptive and exteroceptive elements, necessitating consideration of the different components of feedback during the interaction [7]. Proprioception refers to the internal sense of body position and movement, enabling individuals to perceive the location and orientation of their body parts [66, 91]. In the context of gestural interactions, proprioceptive feedback plays a crucial role in guiding users to accurately reach the desired gesture locations on the back of the hand. On the other hand, exteroception involves external cues and sensory information derived from the environment [91]. In our study, exteroceptive feedback aids in verifying the accuracy of touch locations on the back of the hand. By integrating both proprioceptive and exteroceptive elements, the RadarHand gestures can provide users with comprehensive feedback, thereby enhancing the overall usability and effectiveness of gestural interactions, particularly in the landmarks with the least proprioceptive error. Hence, RadarHand gestures performed around these landmarks exhibit high accuracy during everyday smartwatch interactions under conditions of high cognitive load and without visual cues.

5 STUDY 2: DEEP LEARNING AND GESTURE GROUPING

We established the best gestures based on proprioception and survey results from Study 1. In this section, we aim to design and train a deep neural network model to recognize and classify these touch-based proprioceptive gestures accurately. As we plan to establish a design guideline for RadarHand, we will collect an adequate amount of gesture performance data and train different

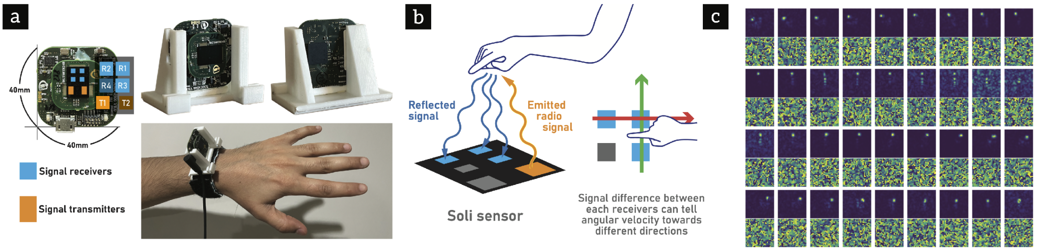


Fig. 7. (a) The Soli sensor hardware with antenna placement introduction (highlighted ones are the antennas being used in prototype system), (b) Soli signal transmission and reception, (c) the range-Doppler data collected from a single signal receiver of Soli (tap gesture, center 36 frames).

models for a wide variety of gesture groupings and combinations, as well as suggest which gesture group is contextually more suited to a proposed RadarHand application.

5.1 Apparatus

5.1.1 Hardware. The entire sensing paradigm for RadarHand depends on the Soli sensor that was mounted on the wrist of the user, and the signal processing pipeline runs as a part of application software. For this trial, the sensor was tethered to a laptop computer for power and data transmission. The sensor circuit board was mounted on a wrist using a custom 3D printed mount with a bangle made of a double-sided Velcro strip so that it was easily wearable. The mount had a channel that allowed a material sample of two millimetres thick to be mounted in front of the sensor. This was to simulate the use of different kinds of hardware casing materials or the sensor surface obscured by clothing. The device is shown in Figure 7. The Soli sensor we deployed was a custom design based on Infineon's BGT60TR24B chip [59].

Soli is a miniature FMCW (Frequency-Modulated Continuous Wave) radar system. Generally, FMCW radar emits a sinusoidal electromagnetic wave and reads the reflected signal after bouncing off an object (Figure 7). The electromagnetic wave changes its oscillation frequency within a specific frequency band available in the system, continuously between lower to higher over time. That allows Soli to measure the distance and velocity of a target object, hence it is possible to extract spatiotemporal information of it.

The data we obtained from the sensor was a time-series complex range-Doppler (CRD) map [29]. The range-Doppler map is a two-dimensional representation of reflected signals captured with a receiver. The range dimension and Doppler dimension are responsible for the distance to an object and the velocity of an object from the sensor surface, respectively. In other words, the range-Doppler map exhibits spatial energy intensity. The CRD map is data represented in complex numbers, which allows us to reconstruct an absolute range-Doppler map that illustrates an amount of reflected energy in the sensing area, and a map of the arrival angle of reflected signals, used to retrieve the angle of the incoming signal to understand angular motion of an object by combining multiple channel inputs. In this article, we call these two maps *the magnitude map* and *the phase map* from the CRD map, respectively.

The sensor device we used equips two signal transmitter antennas and four signal receiver antennas. Since the receiver antennas are arranged to form a square, we can take two pairs of receivers to form an L shape so that we can extract the angular motion of the objects in Soli's sensing field about the azimuth and elevation direction (Figure 7). Thus, we can elicit three key characteristics to read micro gestures with CRD maps: distance, velocity, and angular motion of a finger on the back of another hand [29].

We used the C++-based Soli **software development kit (SDK)**, which has digital signal processing features to elicit adequate features from the raw sensor signal to talk with the sensor and extract CRD map data. As before, all apparatus was sanitized before and after each participant's session.

5.1.2 Software. We employed a 15-inch MacBook Pro laptop (Apple Inc.) for data collection with a Soli sensor device tethered by a USB cable. Software to talk to the sensor for data acquisition, processing, and recording was developed with the openFrameworks software library. In the RadarHand prototype, the sensor was configured to sample signals from three signal receivers at 1,000 Hz. Then, the signal captured was down-sampled with a sliding window of 32 consecutive sample frames. This operation technically realizes low-pass filtering on the sample frames to eliminate noises and avoid confusion in sample frames apart from the hand performing a gesture. After the downsampling, the software streams the data at 32 frames per second. From each sample, we extracted a CRD map composed of three magnitude and phase maps from the receivers. Figure 7 shows an example of consecutive CRD maps captured with Soli while one of the authors was performing a gesture.

5.2 Procedure

The study participant was seated in a shared space in a laboratory building and given an information sheet and a consent form to read and sign. A Soli sensor was put on their left wrist, and the experimenter explained the tasks. Once the data collection started, the participant was shown the gesture to perform with a color image of the left hand with gesture instructions shown in simple graphics on the laptop screen. They had 5 seconds to confirm the gesture to perform from the image. Then, they were prompted to start performing the gesture ten times. A beep sound was played at the start of each gesture iteration to notify the participant when to perform the gesture, along with a progress bar showing each gesture iteration on the laptop screen. Participants had 2 seconds to complete each gesture.

In total, there were 55 gestures plus 1 background noise class consisting of all background data as shown in Figure 9. A combined class of background data is intended to reduce false positives in the prediction and increase the robustness of the model in real-time. The background data was selected based on everyday activities such as clapping, waving your hands, or typing on a keyboard. The complete gesture set was repeated four times with different sensor covering situations with different materials: no material, glass, plastic, and fabric. Each covering material was placed in front of the sensor. Performing the entire gesture set took under half an hour, and the whole session took two hours. The gestures the participant was asked to perform were tapping, vertical slide from the CS to MCP and MCP to DP, horizontal slide from one end of the back of the left hand to another end, and knob rotation.

5.3 Data Collection

We recruited 46 participants (24 females, mean age: 24 years, SD: 2.53). All the participants were right-handed, meaning they wore a Soli on their left forearm and used their right hand for gesture trials. Prior to the main session, we explained to the participants the details of the study and gave them 5 minutes to try out each gesture and software used for collecting data. The participant then watched the laptop display and performed the gesture displayed on the screen. Figure 8 shows the RadarHand gesture samples and Figure 9 shows the study setup from a participant's point of view.

To increase the robustness of the model, we prepared three different material samples to add some variation to the sensor output (glass panel, cloth, and plastic panel, all of them of

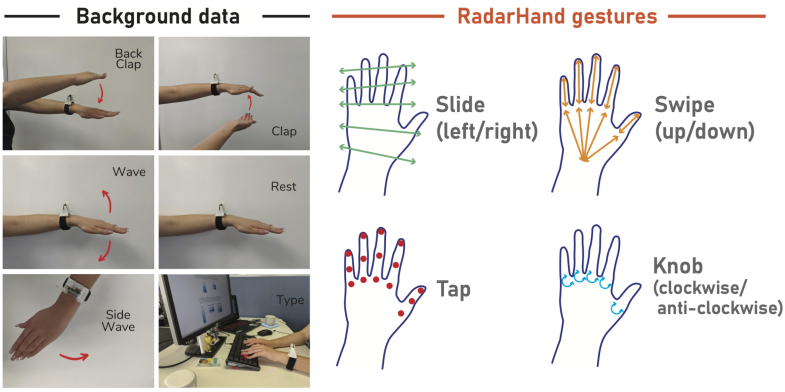


Fig. 8. RadarHand gesture samples.

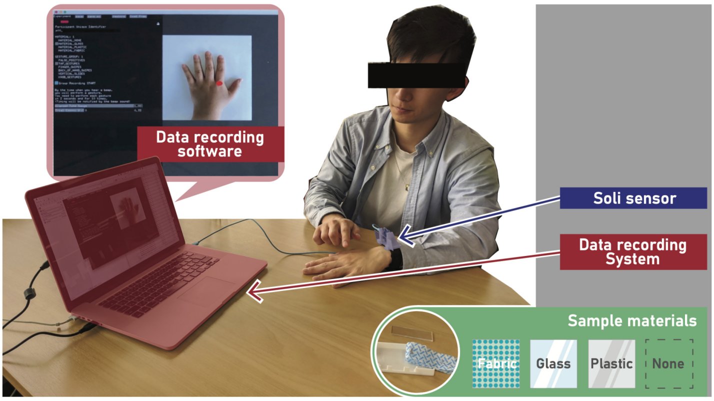


Fig. 9. The data collection setup for Study 2. A participant wears the Soli sensor device facing towards their left hand on their left wrist and performs gestures that is prompted on the laptop computer screen.

two-millimetre thickness). These materials were installed to the channel of the sensor mount to simulate a sensor under different conditions.

In this study, we had four different material conditions along with no material setup to obscure the sensor surface, so we collected 2,440 trials (10 trials * 55 + 6 (55 gestures with 6 variations of background noise) * 4 (repeat)) from each participant, and each data collection process per participant took 2 hours. At the end of the study, we compensated participants with a $40 shopping voucher. Overall, we collected a total of 112,240 samples.

We randomly split each participant's data into training, validation, and test sets. A total of 36 participants' data went to training (1,000 samples per gesture), 6 participants' data went to Validation (200 samples per gesture), and 4 participants' data went to test data (120 samples per gesture). We made sure that gestures from any participant were not split into the training, validation, and testing sets.

5.4 Algorithm

For gesture recognition, we used the Deep RadarNet architecture, which is a modified version of the RadarNet software [29]. As Soli samples at 1,000 Hz, we used a sliding window of 32 frames to

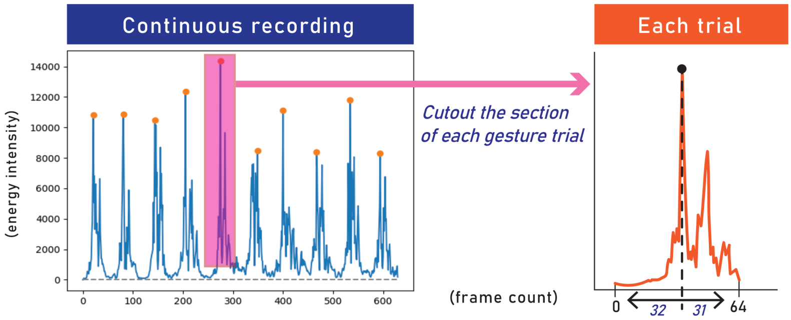


Fig. 10. Ten continuous gesture samples are split into each gesture trial based on a peak detection algorithm. The graph above depicts the energy of the signal of each gesture type.

generate a range-Doppler map to effectively down-sample it to 32 frames per second. This resulted in 64 frames of data over two seconds of recording, which we used as the final time window per gesture. We specifically fine-tuned the model to perform better for close-range gestures like what we proposed in Study 1. The model consists of (1) radar signal preprocessing, (2) Deep RadarNet, and (3) gesture debouncer.

5.4.1 Data Preprocessing. After we split the data into three sets of trials, we refined the timings of the labelled segments attached in the positive recordings. During the gesture recordings, it was observed that there was a timing difference between when gesture prompts were given to participants and when those participants actually performed the gestures. To design a deep learning model that is real-time gesture inference ready, we fed continuous data of each gesture (a recording of consecutive 10 gesture trials) into our pre-processing algorithm. The pre-processing algorithm calculated a time-series plot of maximum energy value from the magnitude component in the range-Doppler map for each frame in the continuous data recording, and found out the peaks using the peak detection algorithm and extracted 32 frames before the peak and 31 frames after the peak, as shown in Figure 10. Once the data was extracted, it was saved in the NumPy[11] data format.

We then generated negative samples from background recordings by extracting 64 frames and splitting them into the training, development, and test sets. With the Soli SDK, we extracted CRD maps for each receiver channel. Each CRD map was composed of 2D data that has 32 range bins and 32 Doppler bins. All of the values were stored as a complex floating-point value and can be decomposed into (1) a magnitude part, which holds information on energy distribution in the sensing area, and (2) a phase part, which contains information on the angular velocity of the object in the sensing area. With the configuration we adopted in this study, we captured CRD maps from three receivers in the sensor, and thus we got three CRD maps per frame (32 * 32-pixel map for both magnitude and phasic parts for three channels).

Deep RadarNet uses complex range-Doppler maps as input, similar to the RadarNet. The Deep RadarNet used the first 16 range bins as an input to detect gestures in a 40-cm radius from the sensor surface. Each bin size can be calculated by the equation:

$$Range(m) = C(speedoflight, 3.0 * 10^8 m/s)/2 * BW(inHz) \tag{3}$$

In this study, the radar signal bandwidth was $63GHz - 57GHz = 6GHz$, and each range bin size could be calculated as 2.5 cm from the equation. We limited the interaction space to 40 cm from the sensor, so we cropped the range bins at the 16th bin, corresponding to 40 cm distance to have

[11] https://numpy.org/

users' index fingers always within the interaction space on CRD maps when their hand performed the gestures. The input was reshaped into a tensor with sizes (16, 32, 6) and passed to the frame model of the Deep RadarNet.

5.4.2 Deep RadarNet. Deep RadarNet takes a CRD map consisting of complex values from three channels as input. In these data representations, the relative phases among the channels correspond to the angle of scattering surfaces around a Soli sensor. The absolute phases of the complex values are affected by many factors, including surface positions under the range bin resolutions, phase noises, and errors in sampling timings, and can be regarded as uniformly distributed random values. While the absolute phases can distribute uniformly in a large dataset, there could be biases in a smaller dataset. To address the potential biases, we augmented the data by rotating the complex values with random phase values chosen from a uniform distribution between $-\pi$ and π. Furthermore, the magnitude of complex values are affected by many factors to some extent, including antenna properties differences among multiple Soli sensors, signal reflectivity of scattering surfaces, and orientations of the surfaces. Thus, we augmented the data by scaling the magnitude with a scaling factor chosen from a normal distribution. These data augmentations can be written as:

$$CRD(r, d, c) = s * CRD(r, d, c)(\cos\theta + i\sin\theta) \tag{4}$$

where: CRD is a complex range-Doppler map; r, d, and c are a range bin index, a Doppler bin index, and a channel index in the complex range-Doppler map, respectively; s is a scaling factor chosen from a normal distribution with a mean of 1; θ is a random rotation phase chosen from a uniform distribution between $-\pi$ and π. While we used the proposed data augmentation technique from the Deep RadarNet research only for our gesture detection system, we believe that this technique can be applied to other deep neural network systems using radar signals as inputs.

The Deep RadarNet consists of a frame model and a temporal model as shown in Figure 11. At the beginning of the frame model, an average pooling is applied to the input tensor to make the tensor size smaller and to reduce the computational cost. After this layer, a series of separable 2D residual blocks and max pooling are applied. We opted to use separable convolution layers instead of standard convolution layers in the residual blocks to optimize computations at the cost of minor degradation in the gesture recognition performance. Finally, one separable convolution layer is applied to compress each channel into one value, shrinking the tensor to 36 values as an output from the frame model.

The temporal model takes the last 64 frames of the frame model outputs as input. The 64 frames of the frame model outputs, except the latest one, were taken from the cache. The temporal model concatenates and processes them with a series of one-dimensional residual blocks and one-dimensional max pooling layers. At the end of the temporal model, a dense layer outputs the classes' variable values, and a softmax layer is applied to compute the probabilities for the variable classes. We found that using an LSTM layer instead of the series of the one-dimensional residual blocks and the one-dimensional max pooling gave performance improvements. However, we opted to use the current structure to reduce computational cost.

5.4.3 Gesture Debouncer. A window size of 64 frames was used for the segmented classification task. However, the algorithm needed to correctly classify the gestures from an unsegmented data stream. This is significantly more challenging since we do not know where the gesture is situated within the time series data. To overcome this, we added the following heuristics: (1) the chances of a gesture to be performed should be higher than an experimentally determined threshold of 0.3 within the last three consecutive frames, and (2) after a gesture is detected, the chance of any subsequent gesture to be detected becomes lower than 0.3 before gesture detection is initiated

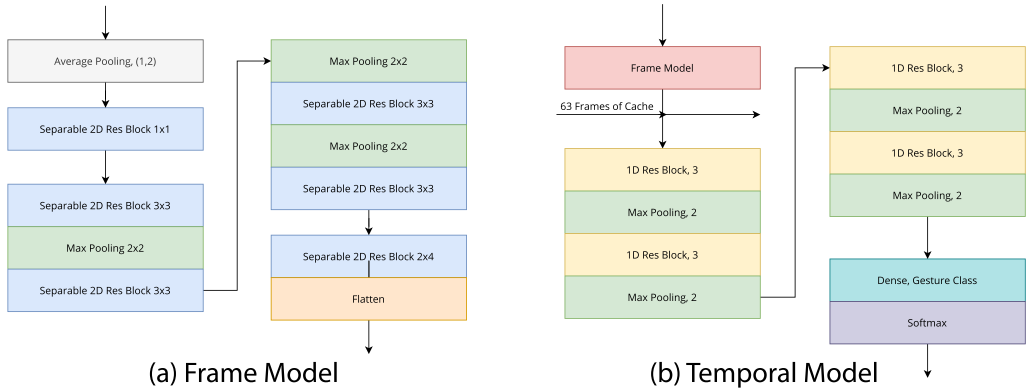


Fig. 11. Deep RadarNet consists of a temporal model and a frame model. The frame model summarizes one frame of complex range-Doppler maps into 36 values. The temporal model combines summaries from 64 frames, then applies a dense layer that outputs gesture class probabilities with a softmax layer.

again. The thresholds were determined experimentally in order to achieve the desired balance between recall and false positives.

5.5 Evaluation

All training was done for 1,000 steps with a batch size of 32. Each training took 5 hours with an NVIDIA V-100 with 32 GB graphic card memory. We evaluated the performance of our architecture in two tasks. The first is a segmented classification task. In this task, recordings in the test set were segmented into multiple samples using the same algorithm we used to generate the training samples. As a result, the samples were easier to classify. For instance, complete gesture motions are always detected at similar positions in the segments. The second task is an unsegmented recognition task, where a model combined with a gesture debouncer had to detect gestures with continuous time-series data without knowing the positions and the numbers of gestures in the data. This made the unsegmented recognition task more challenging. However, algorithms have to process continuous data in practice; evaluations with this task gave us more ecologically valid performance estimates.

According to Study 1 results, the hand fingers are ranked as the thumb, index finger, middle finger, ring finger, and pinky finger based on their least proprioceptive error on tapping abilities. Additionally, we found the ranking of gestures from participants as listed in Table 3 from our survey. Combining the results of Study 1 and the survey, We examined our Deep RadarNet algorithm and proposed 29 gesture groups. We reported each gesture group's test accuracy and its contextual applications.

5.5.1 Computational Efficiency. We evaluated the computational efficiency of the Deep RadarNet using its model size and inference time. The model size affects how much memory is required to run the model. Inference time affects how much computational power a processor needs to run the model, as well as the power consumption and thermal effects of the computation.

We compared with RadarNet [29] and modified its input based on RadarHand's gestural input size. We evaluated inference time using the TensorFlow Lite [1] and its performance profiler distributed with TensorFlow v.2.4.1. We converted all models into TFLite models and measured the inference time on Pixel 4 XL with the performance profiler by taking the average inference times over 1,000 inference trials.

Table 3. Comparison of Computational Efficiency in Terms of Model Sizes and Inference Times

Model	Accuracy[%]	Model Size[MB]	Inference Time[ms]
RadarNet [29]	87.5	1.1	0.57
Deep RadarNet	93.7	1. 9	0.45

Deep RadarNet was as computationally efficient as the RadarNet.

As shown in Table 3, Deep RadarNet has a 3.5 times larger model size and 1.25 times longer inference time than RadarNet. This is because RadarNet has a smaller interaction area compared to Deep RadarNet. Compared with the modified RadarNet with the same interaction area as Deep RadarNet, the model size and the inference time were comparable, indicating that their computational efficiencies are similar. Furthermore, Deep RadarNet achieved higher classification accuracy than RadarNet when it comes to detecting gestures set of 8. In addition, RadarNet [29] have proved to be efficient compared to previous works baseline models [31, 73]. These comparisons demonstrate that Deep RadarNet is as computationally efficient as RadarNet and can be executed on mobile devices with limited computing resources.

5.5.2 Segmented Classification Task. We generated a segmented test set for each gesture trial, along with the training samples from the training set that were generated with a sample generation algorithm. Each sample had 64 frames of CRD maps from three receivers and was assigned one of the classes as a ground truth label. Then we evaluated the classification result of all gesture trials in the segmented test set using Deep RadarNet. The labels with the highest probability among the gesture classes were used as predictions without applying the gesture debouncer. This was because the test recordings were pre-segmented into samples in the segmented classification task. Figure 13 shows the results, which will be explained in the Gesture Grouping and Accuracy Results section, demonstrating that Deep RadarNet provided robust classification performance in the segmented classification task.

5.5.3 Trial on Applying Different Frame Size. As each gesture was observed at any time in a two-second time window, with 64 frames worth of data, we tried to find the best-performing frame size by trying a few different frame size conditions. To examine that, we took 8 unique gestures based on the Study 1 results, which gave us the highest accuracy from the Deep RadarNet inference trial. Then, we fed this as input to the Deep RadarNet algorithm. One of the strong motivations of this trial is that, as we mentioned in the previous section, we observed some gestures starting timing drift due to each participant's response to the beep sound and gesture speed variability. Hence, it is worthwhile to know the best-performing frame size as an input to the model, which is directly proportional to the parameter size and computational power required for inference. We compared all the conditions we examined using the results of the development set. Figure 12 shows the graph of the validation accuracies for different input frame sizes. We also tried a sliding window of 30 consecutive frames across 64 frames of data under the same labelling as input to the deep learning model. Even though a window size of 56 yielded the highest accuracy, we opted for 64 instead, which has a lower accuracy by 1, yet can better account for different gesture timing across participants.

5.6 Gesture Grouping and Accuracy Results

In this section, we design groupings for our gesture combination and report their accuracy. We group the gestures for 2 reasons. First, it is unrealistic and redundant for us to report inference

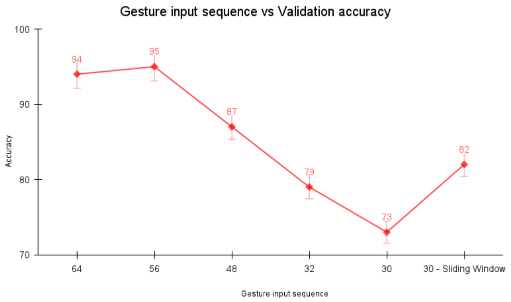


Fig. 12. Validation accuracy for gesture input with different frame sizes.

accuracy for every possible gesture combination. Second, we wish to design gesture sets that cater to specific contexts of applications. To group the gestures, we consider several aspects listed below.

Each group was trained with an additional class for background noise, hence three gestures actually mean four classes, and so on:

(1) Gestures were selected based on the gathered results of Study 1. The thumb, index, and middle finger are the least proprioceptive error and are preferred areas for gestural interaction. MCP and DP were also found to be the least proprioceptive error and preferred spots. Additionally, one of the limitations was the defined orientation. If the orientation changes, the results will also change. Thus, we include all fingers in our gesture combination trials.
(2) We referenced **smartwatch user interfaces (SMUI)** when narrowing and grouping the gesture selections [16].
(3) We selected three gestures, including tapping, sliding, and rotation gestures.
(4) As the group size increased, we narrowed the gesture selection to models that had previously performed well.
(5) We generally avoided placing similar gestures adjacent to each other and at the same location since even a slight change in sensor placement on the wrist can have a significant effect on its performance.
(6) Since we do not calibrate for hand size, we limited the gestures per region to only three to minimize false positives. All of the gesture set results are illustrated in Figure 13.

The grouping of various gestures enabled gesture interaction to take on a new dimension by distinguishing between gestures for essential interactions, such as answering/dismissing a phone call, and gestures that require a great deal of attention, such as increasing the brightness of a smartwatch. We created a gesture group consisting of 3, 4, 6, 8, and 10 gestures. Additionally, we created a generic gesture combination trial that combines all the basic gestures regardless of the position of the user's fingers. We hypothesize generic gesture combination trial will be a robust model and can be integrated with any situation. The grouping below provides the rationale behind the grouping as well as its potential application in SMUI. The confusion matrices for all the trials are attached in the Appendix for reference.

3 Gestures (Sets 1 to 6). Referring to Table 3, we focused on the three least proprioceptive error fingers, which are the thumb, index, and middle finger, in that particular order. For the points on each finger, we also focused on DP for the least proprioceptive error landmark, followed by MCP from user preference. As the DP of the thumb is the least proprioceptive error landmark, we envision its use for something crucial, such as an emergency button, and maintain this throughout all the groups. We disregarded direction as of this moment, focusing only on vertical up-slide and clockwise rotation. A total of 6 gesture sets were derived from this, where sets 1 to 4 were gestures

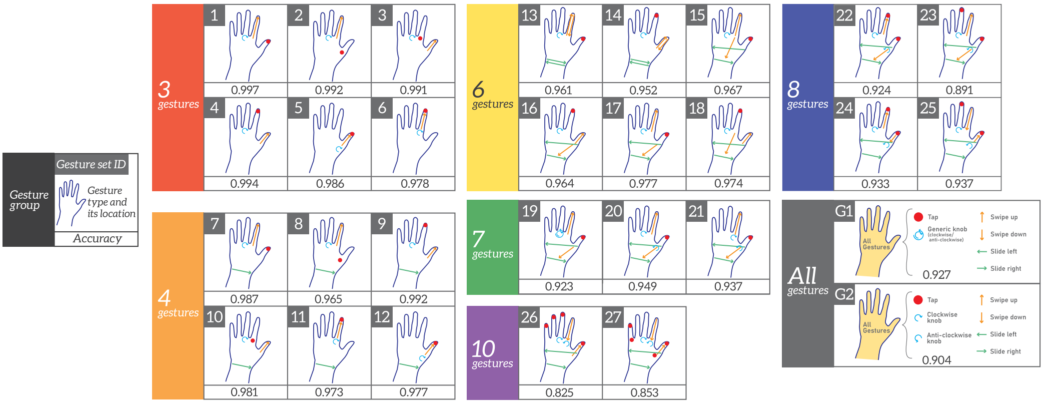


Fig. 13. Inference results on different gesture groups. Inference accuracy is better when the accuracy value is closer to 1.

between the three fingers, and sets 5 and 6 focused on only the thumb and index fingers. For sets 1 to 4, the thumb and index were reserved mainly for tap and slide for being overall more preferred, as shown in Table 3. Therefore, rotation was left mainly on the middle MCP. This overall grouping is suitable for the most basic SMUI like the Fitbit Alta,[12] with a single tap for selection, single slide for menu interface, and single rotation to imitate a one-directional knob.

4 Gestures (Sets 7 to 12). For this group, we added the horizontal right slide gesture to the previous group, making a total of 4 gestures. We chose the right slide as it mimics gestures like "slide-to-unlock". From Table 3, we chose the gesture only to be performed on CS. It can be seen that sets 5 and 6 from the last group, as well as sets 11 and 12 have a dip in accuracy, which leads us to focus on spreading the gestures across different fingers for future groups. This grouping is suitable for basic SMUIs as well, with an additional function to perform a slide-to-delete or slide-to-unlock.

6 Gestures (Sets 13 to 18). We skipped the group for 5 gestures because we decided to directly expand the slide gesture to cover both directions (up and down, as well as left and right). Additionally, we selected the 2 best results from the previous group (sets 7 and 9) and expanded on them. Set 10 was removed because even though it performed well previously, its gestures have high proprioceptive error compared to 7 (slide was performed on thumb and tap was performed on the index) and are less accurate compared to 9. Sets 13 and 14 were based on 7 and 9, respectively. Set 15 is a variation of 13, whereas 16 is a variation of 14 by shifting the down vertical slide to ML of the respective finger. Set 17 is a variation of 15, and 18 is a variation of 16, where the vertical slides have experimented between the index and thumb since we found from the previous group that variation among fingers provides better results. This grouping is suitable for a fully functional SMUI with a knob, which is close to a watch crown-inspired interface, such as the Digital Crown on Apple Watch.[13]

7 Gestures (Sets 19 to 21). For this group, we again chose the best option from the last, namely set 17 (set 18 was deemed less proprioceptive since the tap gesture was on the index). Additionally, we added the counter-clockwise rotation gesture for a more sophisticated virtual knob control. Set 19 places both rotations on the middle MCP, whereas 20 and 21 split them between the thumb, middle, and index. This grouping is suitable for SMUIs with larger scale knob interface, such as the Samsung Galaxy Watch 4[14] with a rotational bezel as a key SMUI navigation component.

[12] https://www.fitbit.com/gb/shop/altahr
[13] https://www.apple.com/watch/
[14] https://www.samsung.com/global/galaxy/galaxy-watch4-classic/

8 Gestures (Sets 22 to 25). With all the basic gestures and directions added, the next step for us would be to simply add multiple interaction points of the same gesture type. Since tap was overall preferred, we added one more tap gesture based on the previous best-performing set, which is set 20. Sets 22 and 23 replicate set 20 with the added tap on the index DP and middle DP, respectively. Set 24 is a variation of 21 by shifting the down vertical slide from the thumb ML to the finger, and set 25 is a variation of 22 that does the same.

10 Gestures (Sets 26 and 27). Here, we further push the limits of the tapping gesture by doubling it to 4 taps, hence skipping 9 gestures. Sets 26 and 27 are variations of the previous sets 24 and 25 that performed the best. Here, we experimented with including the high proprioceptive error landmarks like the pinky and ring finger. This was for two reasons: (1) we wished to split the tap point according to actual distance to resemble how the buttons on a watch typically look, such as 2 buttons on each side, or 3 on one side, and 1 on the other, and (2) the models will perform better in general when the tap locations are physically distinct from our initial testing. The thumb MCP knob rotation has also been changed to the index MCP to reduce the number of gestures per finger.

Generic Gestures (G1 and G2). For this group, we disregard the location of the gestures and simply focus on generalizing all 7 gestures. For example, tapping on the thumb DP is the same as tapping on the index DP. This group does not take into account proprioception errors and allows lower accuracy, making it more suitable for multitasking, such as interaction while driving or cycling. These gesture sets are also independent of hand state, where participants can perform the gesture when their hands are busy. A similar interface is available in the navigation mode of smartphone's music application, which have larger UI components for ease of maneuver while driving. Additionally, it can be used in everyday smartwatch interaction scenarios and act as a modeless interface.

6 STUDY 3: REAL-TIME EVALUATION

In this section, we aim to evaluate our model's performance in real-time using new test datasets with real-world noise and determine false positives in the real-world scenario. For real-time evaluation, we chose the discrete gesture set 25, which includes all the eight gesture possibilities, and the generic gesture set $G1$, which gave the highest segmented classification accuracy. The generic gesture set $G1$ is not location-specific. Hence, we hypothesize that, in real-time, the gesture inference will have more false positives compared to the discrete gesture set of 25. This is because there is more chance of free human movements similar to gestures which could produce false positives under real-time sensing in a real-world scenario.

6.1 Study Design

Besides selecting the discrete and generic gesture set, we further break both models down to with and without background classification.

We do this because we believe smartwatches conventionally function in two modes: (1) active interaction, where the user initiates the input to achieve a desired output, such as tapping a button to access an app intentionally, and (2) reactive interaction, where the device initiates interaction and requires the user to react to it, such as replying to a notification prompt. Active interactions require background classification since it is always actively aware of user input, though this consumes more background computational resources and power. Reactive interaction, on the other hand, does not require background classification since it only listens after the prompt. This makes it less intuitive, yet overall consumes less power and is suitable for better battery life.

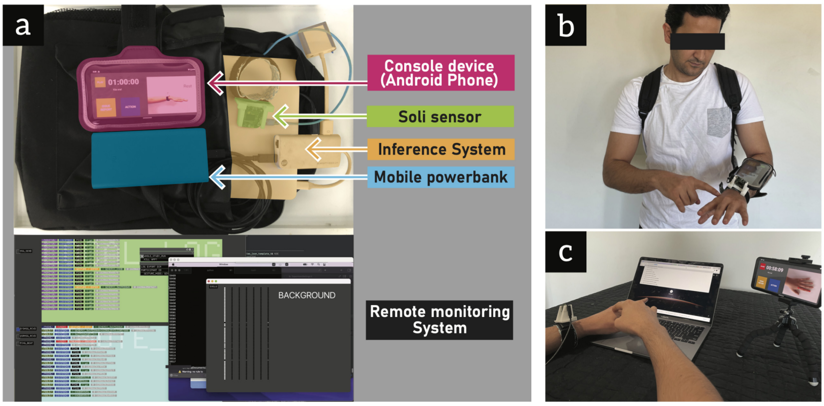


Fig. 14. Real-time study setup for Study 3. (a) Apparatus will sit in a backpack while a participant works on the study except for the Soli sensor, and a remote monitoring system will help the experimenter conduct the session, (b) shows a participant performing a gesture in the standing pose, and (c) a participant performing a gesture on the desktop.

6.2 Apparatus

To conduct this study, we developed three systems; a backpack, an on-wrist wearable, and a remote monitoring system. The backpack is a self-contained gesture inference system using Soli. It equips a laptop computer (Apple 13-inch MacBook Air), a portable battery, and a dummy external display adapter. The laptop ran a live gesture inference with a tethered Soli sensor, while the external battery powers the laptop for the stable system performance of the current prototype. The dummy display adapter is to make the inference system keep running while the laptop is folded in a backpack. The tethered Soli sensor is on the left wrist as a part of the on-wrist wearable system.

Alongside the Soli sensor, we adapted an Android smartphone (Google Pixel 4 XL) as a participant's communication terminal during the study session. A custom Android application acted as a console for the participant to log their gestures and report their current activity or issues with the system. The wearable (smartphone and Soli sensor) was worn on the participant's left wrist.

The remote monitoring setup gathers all the log data related to the activity of a participant or study status transmitted from these two systems. The response made by a participant on the on-wrist wearable or gesture inference status from the backpack will be sent to the C++-based monitoring software that runs on a 15-inch MacBook Pro laptop via the same Wi-Fi network across all these three devices. This application also allowed the experimenter to manage the study and data logs. Each of these devices was using the WebSocket protocol to send data back and forth between systems, and a simple cloud server runs on Amazon Web Services[15] has been used to assist the communication. We illustrate the full setup in Figure 14(a). All the apparatus was properly sanitized before and after each participant.

6.3 Study Procedure

At the start of the study session, the experimenter introduced the tasks the participant would perform and the apparatus they would use. After the briefing, the participant was asked to wear an equipped backpack and the wearable hardware prior to the study session. They could wear a Soli sensor in the same manner in Study 2 and a smartphone with an arm mounting belt. Figure 14(b)

[15] https://aws.amazon.com/

shows an example where a participant performs a gesture with the system setup. The experimenter monitored the system status and the participant's actions via the activity logs throughout the session. All sessions were taking one hour approximately.

The participant was asked to perform the gestures while they performed daily activities in an office space, including working with a computer on a desk, walking, standing, chatting, or cooking in a small kitchen for brewing tea or coffee.

The monitoring application was designed to prompt the participant to perform the next gesture automatically through the console on their left arm for each gesture trial. The timing was notified by the vibration of the console and an image shown on the smartphone screen. The participant tapped the image to start a gesture performance and tapped again to finish it. The participant was asked to report an action they did at the time from the console after each trial by selecting a predefined action or describing the action by using an on-screen keyboard. Each gesture was repeated three times throughout the study. The order of the gestures performed and the interval between the gesture trials were randomized from 30 seconds to 4 minutes for each participant.

The participant was asked to report all incorrect gestures they performed during the gesture prompts and prior to the gesture prompts in the monitoring app. In addition, when they encountered any trouble or issues around the study, they reported it to the examiner through the console. An examiner could also respond to the participant through the monitor software if the participant needs assistance. For each condition, it took one hour to conduct the whole session. At the end of the study, we compensated the participant with a $10 shopping voucher.

6.4 Data Collection and Processing

We recruited 12 participants (6 females, mean: 24.4, SD: 4.0). The criteria for the recruitment of participants were the same as the studies from the previous sections. We asked participants to remove any rings, wristwatches, and other hand-worn objects. All the participants had never experienced the RadarHand gestures previously.

As mentioned in the previous section, gesture sensing with Soli performs with a sampling rate of 1000Hz. In order to reduce the data size for the real-time gesture inference, we applied a sliding window of 32 frames, similar to what was done in Study 2. As a result, we acquired 30 to 32 frames of data for each second from three signal receiver channels. A C++-based software program captured the sensor data, and each frame data was piped to another Python-based inference server program running simultaneously. The Python script infers the gesture in real-time based on incoming frame-by-frame data with the selected models from Study 2 results. The program continuously updates the frame history for the last 64 consecutive frames, including the one recently loaded, and it infers any gesture with the frame history for every two incoming frames based on the gesture debouncer algorithm discussed in Study 2. We ran the inference for every two frames to make sure the system ran the inference in real-time without any delay on inference result delivery. Both of the programs were linked by the ZeroMQ protocol for frame data transporting and inference result reporting.

All the log data was recorded in the CSV file format, and each line shows each data sent from another device at a particular time. We recorded the following: (1) timestamp of the recording device, (2) device name (monitor, inference system, and wearable console), (3) timestamp of the device sending the data, (4) remaining study duration, (5) data owner (system and participant), and (6) any action performed by the participant. The action log contains detailed information on each gesture trial that a participant performed, an action report from a participant, the situation of the study, and the network connection status. The Wi-Fi connection between the devices actually caused some time lag for each data transaction. To overcome this, we collected timestamp information from all the devices and a "ping" signal, which was continuously sent from the main monitor computer to understand how much time drift was needed to accurately process the log data. Each data packet exchanged between the devices came with unique information that was related to a particular device or action. We combined all the data that we collected from those devices to process as one singular dataset for post-analysis. We also developed a simple data processing script program with Python to analyze and plot the data for each gesture trial.

For the data processing, we checked the system log file from the monitor system (observer system), the inference system (the backpack), and the real-time inference history simultaneously. First, we looked at the gesture section from the system log and found the corresponding Soli frame number for the beginning and the ending of the trial. Next, we checked the last 10 ping response times between the monitor system and the wearable console to check the average duration of the delay taken for message sending. Depending on the delay, we added an extra 16 frames of data if the delay was less than half a second and an extra 32 frames if it was more than a half second before the trial section was reported. An extra 32 frames after a trial section was also considered since each gesture inference required 64 frames of data. After that, we extracted a subsection of the inference history within a trial section to elicit the gesture performed. We applied the gesture debouncer from Study 2 to get the gesture performance history out of the last 64 frames.

We applied a model trained on both gesture and background data for active interaction. The model's inference was predicted based on the input data from the sensor for the entire duration of the study, including when the system was prompted to perform a gesture. We used the model that was trained without background class data to predict reactive interaction. The inference of this model could only be predicted when participants were prompted to perform the gesture. The gesture with its confidence rate of the inference was documented as a result for each gesture section.

6.5 Results and Discussion

As shown in Figure 15, the results from the reactive interaction situations showed clear classifications on all gestures compared to the active interaction situations. In the following sections, we will discuss the results and findings from each study condition in more detail.

6.5.1 Active Interaction. The system achieved 87% accuracy as shown in Figure 15(a), with the generic gesture set ($G1$ in Figure 13) and 74% as shown in Figure 15(b) with the discrete gesture set (set 25 in Figure 13).

Since the generic gesture set was trained to be more tolerant towards the location of the gestures, it resulted in overall much fewer false positives and negatives. This may also lead to the gestures being overall more understandable and easier to master. Among the gestures, generic knob (100.0%), generic slide up (94.4%), and generic tap (88.9%) performed well.

The main issue with the generic gesture set is that all gestures have an interaction region overlap. For example, the tap and knob gestures can both be on the index's MCP. This led to several misclassifications, such as the generic tap, slide left, and slide right being occasionally recognized as the generic knob. Among all the gestures, it can be seen that both the generic slide left (77.8%) and slide right (55.6%) performed the poorest. This is possibly due to the presence of different hand states when the participant is performing other tasks. Ideally, they would need to stop their task before performing the gesture, but we did not enforce this to observe the behavioural impact on the model performance. The generic slide right gesture overall performed the worst, possibly due to misrecognizing the initial approach of the right hand towards the left side of the left hand, which can be seen as a tapping or sliding up gesture at any of the regions around the pinky.

For the discrete gesture set, we found that the majority of the misclassification is due to the model recognizing the gesture as background. This is due to various factors like the speed of gesture performance, finger arrival and departure angle, finger length and hand state, which we

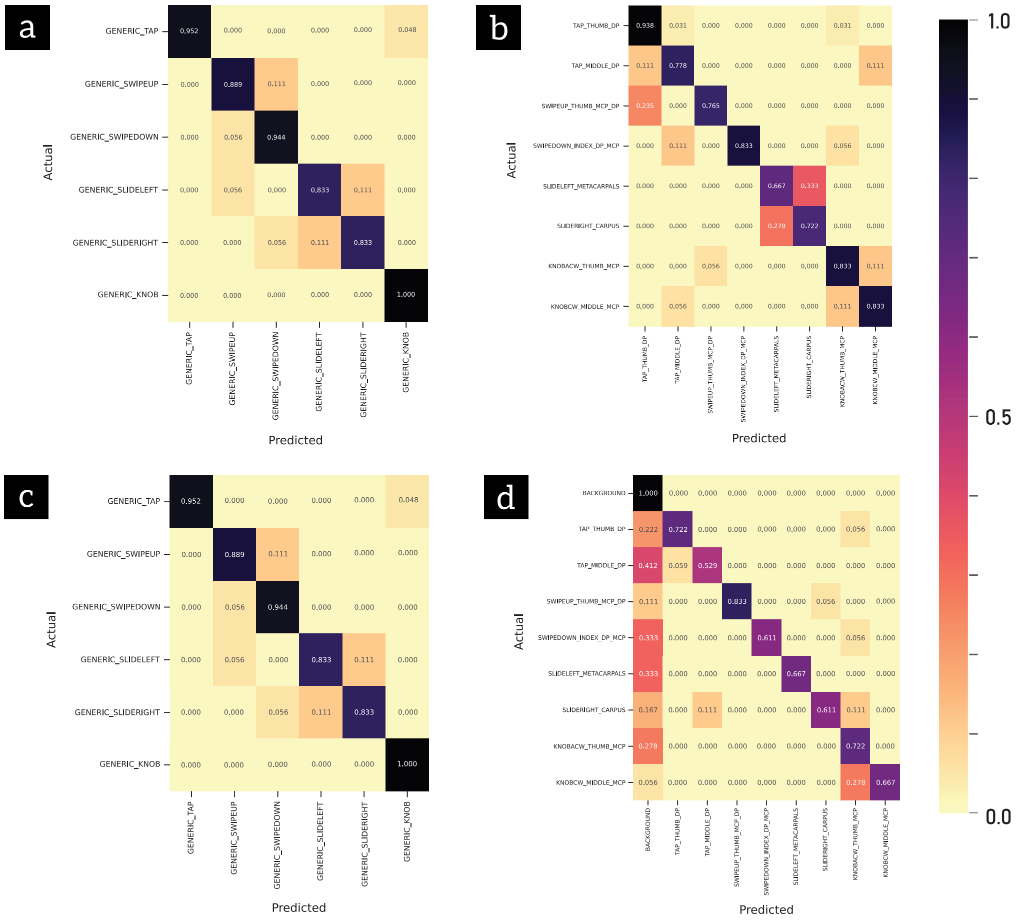


Fig. 15. The confusion matrices for the (a) generic and (b) discrete models for the reactive interaction situations, and (c) generic and (d) discrete models for the active interaction situations.

did not enforce in this study. The worst performer was tap on middle DP (53%), followed by slide down on index DP to MCP (61%) and slide right on CS (61%). All of them tend to be misclassified as background noise. Slide up on thumb MCP to DP (83%), tap on thumb DP (72%), and knob counterclockwise turn on thumb MCP (72%) performed the best possibly due to the thumb being the least proprioceptive error and that it is physically the furthest finger away. We also noticed that the knob clockwise gesture on the middle MCP tends to be misclassified as counter-clockwise being performed on the thumb. This was actually observed as a mistake performed by three participants who performed a counter-clockwise rotation instead.

6.5.2 Reactive Interaction. The system achieved 91.3% accuracy as shown in Figure 15(c), with the generic gesture set (*G*1 in Figure 13) and 81.7% as shown in Figure 15(d) with the discrete gesture set (set 25 in Figure 13). Overall, the performance of forced interaction models performed well in real-time compared to the voluntary interaction models.

The reactive interaction generic tap gesture performed accurately (94.6%) compared to the active interaction generic tap (88.9%). The generic swipe-up (88.9%) and swipe-down (94.4%) have similar movements from the participant while performing the gesture, which causes misrecognition. Similarly, the generic slide right (83.3%) and slide left (83.3%) gesture performance was misclassified due to similar hand movements by the participant while performing the gesture. The generic knob (94.4%) is misclassified as a generic tap due to region overlap between the tap and knob gestures.

For the discrete gesture set, we found that the majority of the misclassification is due to similar gestures, irrespective of their position. The worst performer was slide left metacarpals (66.7%) and slide right carpals (66.7%). Additionally, the swipe-up thumb MCP (76.5%) to DP is classified as tap thumb DP (93.8%) due to the region overlap and similar movement before performing the gesture.

7 DESIGN GUIDELINE AND POTENTIAL APPLICATIONS

Overall, RadarHand performed well in our presented context. As discussed previously, unlike competing sensing methods, radar does not rely on image capture to preserve privacy as an everyday wearable, yet is also able to capture enough spatial information to accommodate the gestures we propose (close proximity finger interactions), unlike IMU-based methods. Its only potential tradeoff is that achieving this performance requires a large and diverse dataset, as we detailed in the Study 2 results section. We summarize the key findings from the three studies into a preliminary design guideline for wearable, radar-based, touch-based, proprioceptive hand gestures:

(1) *Proprioception and Gesture Response Time Decrease from the Thumb to the Pinky Finger.* This was tested even under high cognitive load. From the discussion Section 4.7 of Study 1, we propose that emergency functions are kept on the thumb, while the index and middle fingers can be used for device interaction.

(2) *Tapping Gestures Are Most Preferred, whereas Rotation Gestures Are Least Preferred.* From the discussion Section 4.7 of Study 1, we suggest tapping on the hand could be used more for intuitive and immediate action on the smartwatch applications (e.g., on/off options or app selection). Navigating between screens and applications on a smartwatch can be accomplished through sliding gestures. Additionally, rotational gestures may be used in applications requiring medium to high cognitive load, such as controlling the screen's brightness precisely.

(3) *Choosing Gestures for a Discrete Gesture Set Should Avoid Placing Similar Gestures Near Each Other for Radar-Based Smartwatch Interaction Devices.* This allows the model to be more robust against a slight shift of the device on the wrist, which may cause false positives if similar gestures are nearby. In addition, the user performing similar gestures nearby may perform in the wrong position. For example, suppose the gesture set has knob gestures on the index MCP and middle MCP. There is a high chance that users performing the gesture will get confused about this location in discrete gesture set conditions, which we have noticed in Study 3 results.

(4) *Limit the Gestures for Each Proprioceptive Region of the Hand.* Despite RadarHand being able to detect up to three types of gestures on a single point, we recommend setting no more than three gestures per region to minimize false positives since each region has a physically small interaction space as described in Section 5.6.

(5) *Design Gesture Grouping for Contextual Applications.* In Section 5.6, we discussed gesture sets from three to ten gestures and distinguished between discrete and generic gestures. Referring to the gesture set in Study 2, we propose grouping various gestures into app-specific gestures or gestures that are only known to smartwatches. Each set caters to different SMUIs, such as basic smartwatches, smartwatches with knobs, and so on, and can be referred to in Section 5.6 for future works.

(6) *Collect Data for Radar-Based Gesture Devices with Material Filters to Improve Their Robustness of Prediction in Real-Time.* This filter must have properties similar to device encasing or

clothes that would be covered such as glass, plastics, and cloth [72]. This will enable users to perform gestures even when the smartwatch is occluded.

(7) *Generic Gestures and Discrete Gestures Mainly Differ Based on the Location-Specific Design.* Both discrete gestures and generic gestures can be operation eyes-free. However, discrete gestures are suitable for interfaces that require more selection options, such as multiple buttons and sliders on a mobile operating system. Generic gestures are more suitable with lower attention and selection requirements with more oversized, single buttons and sliders, such as answering a call.

(8) *For Active Interaction. Generic Gesture Sets Are Preferable. For Reactive Interactions, Discrete Gesture Sets Are Preferable.* For active interaction, the smartwatch is always-on and expects input from the user. The model predicting input should be robust and reliable against the user's everyday movements. Therefore, the generic gesture set is more suitable for active interaction. Similarly, for reactive interaction, the smartwatch expects input from the user after its prompt. Therefore, the system does not need to classify background actions. We suggest the discrete gesture set for reactive interactions.

Aside from the SMUIs mentioned above, there are many potential applications for our gesture input technique. These could include:

(1) *An Intuitive Desktop Input System for GUI Navagation.* Current desktop systems use a keyboard and mouse, which relies on shortcuts for professional software like Photoshop.[16] Touchscreens or touch surfaces like Apple's TouchBar[17] can also be used, but the finger can block the content or is simply not eyes-free. We envision that our system could provide a more proprioceptive alternative, as shown in Figure 16(1).

(2) *Combination with Hand Tracking for Novel Interactions.* With current extended reality solutions now offering hand tracking, we envision RadarHand supplementing the interaction with RadarHand gestures, as shown in Figure 16(2). This could also be enhanced with haptic feedback by manual hand gestures [52].

(3) *Eyes-Free Vehicle Navigation System.* It is important to always keep your eyes on the road while driving or cycling. With RadarHand, performing eyes-free gestures becomes much safer, as shown in Figure 16(3). Furthermore, a "Drive Mode" could be introduced for the model to switch to a simpler version, such as only three gesture sets or a generic set.

(4) *Proprioceptive Interfaces for Prosthetic Hands.* We envision that RadarHand is also usable for users of prosthetic hands and may even possibly assist in issues like phantom limb pain, as shown in Figure 16(4). By leveraging proprioception, the user may have a better sense of body ownership, thus improving prosthetic hand postures, although further studies are needed to verify this [60].

(5) *Proprioceptive Interfaces for Gaming.* RadarHand gesture input could be used to control games, allowing players to interact with the game using natural gestures and movements. RadarHand would always be available and eyes-free input for gaming and entertainment applications.

Proprioceptive touch-based gestures could be useful in real-life scenarios where the user is focused on a particular primary task and also need to attend secondary interactions with devices such as smartwatches or phones. For example, while driving the car, if the driver receives a phone call that they need to respond to, their concentration shifts slightly from driving the car to

[16] https://www.adobe.com/products/photoshop.html

[17] https://support.apple.com/en-au/guide/mac-help/mchlbfd5b039/mac

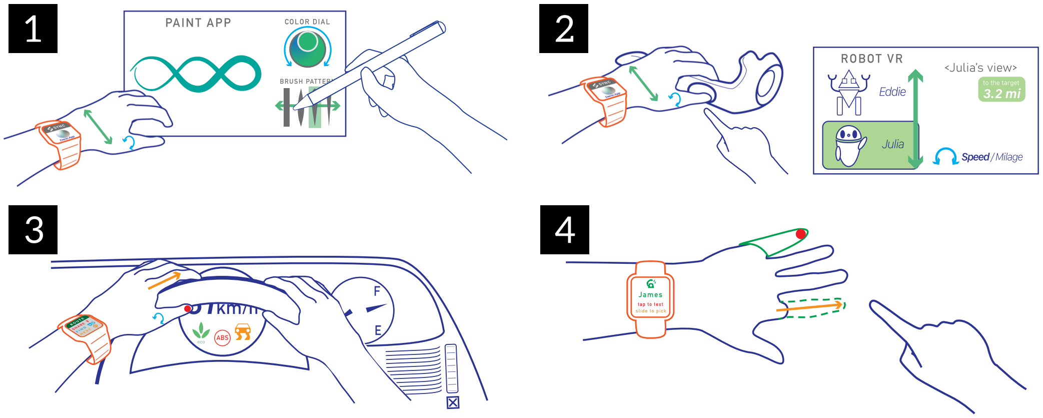


Fig. 16. Potential applications of RadarHand: (1) An assistive desktop input system for conventional GUI-based system, (2) Combination with hand tracking for novel interactions in XR experience, (3) Eyes-free vehicle navigation system, and (4) Proprioceptive interfaces for prosthetic hands.

responding to the call on their smartphone or smartwatch. It may have catastrophic consequences if attention is diverted in this manner. In such instances, users can use discrete gestures to attend to or decline calls while minimizing their cognitive load during the interaction, thus maintaining their attention while driving.

8 LIMITATIONS AND FUTURE WORKS

There are a number of limitations that could be addressed in future work. In Study 1, we only tested gestures on a specific body part at a specific orientation. These two factors can significantly affect the Fitts' law ID since there will be a drastic change in the user's motion and proprioceptive sense for different body parts and orientations. We plan to explore other body parts and different hand orientations to understand this in the future. In addition, we used Fitts' Law to identify or score the best finger tap position to perform gestures in that area. This design may not apply to all hand gestures. We plan to explore this research gap in the future by proposing an extended Fitts' Law to assess proprioception for hand-based gestures.

For Study 2, we collected our data by considering that the participant's hands were placed on the table every time. We did not consider different hand sizes, shapes, or movements of the dominant hand, which may lead to poorer results. Hence, we plan to expand this with various hand states and movements. Additionally, we intend to collect more data combining both background and gesture data from a wide range of participants with a variety of hand sizes in order to improve our model's generalizability and robustness.

For Study 3, the current dataset we obtained does not consider the influence of the gesture performances under different background conditions, hand accessories, and the displacement of the arm wearing a radar sensor. To improve on this study, our next step is to perform a data collection procedure that is closer to real-world use, such as while performing daily tasks like holding a bag, walking, and so on. Furthermore, Our study was designed and conducted on only right-handed participants. In the future, we would like to extend recognising gestures to all participants using few-short or zero-short learning methods [84]. Finally, we designed our gestures for bi-manual interaction with limited types of gestures. In the future, we plan to investigate single-handed interaction and explore different hand gestures and hand-to-object interactions with the aid of proprioception.

9 CONCLUSION

We present RadarHand, a wrist-worn radar for detecting proprioceptive touch-based gestures on human skin landmarks. We first established the most appropriate proprioceptive points on the back of the left hand for gestures. In addition, we grouped and classified the gestures using a deep learning model (with an accuracy of 92% for a generic gesture set and 93% for the best discrete gesture set) and proposed contextual applications for each gesture group. Furthermore, we conducted an additional real-time evaluation of the model to understand its performance and shortcomings based on active and reactive interactions. We obtained an accuracy of 87% and 74% for active generic and discrete gestures, respectively, as well as 91% and 81.7% for reactive generic and discrete gestures, respectively. Finally, we summarized the findings in a set of design guidelines based on the gathered results regarding radar as a wrist-worn wearable. Our results indicate that RadarHand has a lot of potential for future on-skin interfaces.

The data that support the findings of this study are available from the corresponding authors, Ryo Hajika and Tamil Selvan Gunasekaran, upon reasonable request. The data will contain all RadarHand Dataset, Deep learning model weights, software, and results from all the studies. Researchers interested in accessing the data can contact both of the authors to request the data and discuss any necessary terms or restrictions.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Rachel F. Adler and Raquel Benbunan-Fich. 2015. The effects of task difficulty and multitasking on performance. *Interacting with Computers* 27, 4 (2015), 430–439.

[3] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. 2021. Vid2Doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI'21)*. ACM, New York, , Article 292, 10 pages. https://doi.org/10.1145/3411764.3445138

[4] Nuwan T. Attygalle, Luis A. Leiva, Matjaž Kljun, Christian Sandor, Alexander Plopski, Hirokazu Kato, and Klen Čopič Pucihar. 2021. No interface, no problem: Gesture recognition on physical objects using radar sensing. *Sensors* 21, 17 (2021). https://doi.org/10.3390/s21175771

[5] H Charlton Bastian. 1887. The "muscular sense"; its nature and cortical localisation. *Brain* 10, 1 (1887), 1–89.

[6] Kamil Behún, Alena Pavelková, and Adam Herout. 2015. Implicit hand gestures in aeronautics cockpit as a cue for crew state and workload inference. In *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 632–637.

[7] Joanna Bergström and Kasper Hornbæk. 2019. Human–Computer interaction on the skin. *ACM Computing Surveys (CSUR)* 52, 4 (2019), 1–14.

[8] Joanna Bergstrom-Lehtovirta, Sebastian Boring, and Kasper Hornbæk. 2017. Placing and recalling virtual items on the skin. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1497–1507.

[9] Idil Bostan, Oğuz Turan Buruk, Mert Canat, Mustafa Ozan Tezcan, Celalettin Yurdakul, Tilbe Göksun, and Oğuzhan Özcan. 2017. Hands as a controller: User preferences for hand specific on-skin gestures. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1123–1134.

[10] Eric Burns, Sharif Razzaque, Abigail T. Panter, Mary C. Whitton, Matthew R. McCallus, and Frederick P. Brooks. 2005. The hand is slower than the eye: A quantitative exploration of visual dominance over proprioception. In *Proceedings of the IEEE Conference on Virtual Reality 2005 (VR 2005)*. IEEE, 3–10.

[11] Yeonjoo Cha and Rohae Myung. 2013. Extended Fitts' law for 3D pointing tasks using 3D target arrangements. *International Journal of Industrial Ergonomics* 43, 4 (2013), 350 – 355.

[12] 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*. 3403–3414.

[13] 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 *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom'19)*. ACM, New York, , Article 17, 16 pages. https://doi.org/10.1145/3300061.3300124

[14] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST'14)*. ACM, New York, , 161–166. https://doi.org/10.1145/2642918.2647396

[15] Travis Deyle, Szabolcs Palinko, Erika Shehan Poole, and Thad Starner. 2007. Hambone: A bio-acoustic gesture interface. In *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*. 3–10. https://doi.org/10.1109/ISWC.2007.4373768

[16] Tilman Dingler, Rufat Rzayev, Alireza Sahami Shirazi, and Niels Henze. 2018. Designing consistent gestures across device types: Eliciting RSVP controls for phone, watch, and glasses. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI'18)*. ACM, New York, , 1–12. https://doi.org/10.1145/3173574.3173993

[17] Gustav Theodor Fechner. 1860. *Elemente Der Psychophysik*. Vol. 2. Breitkopf u. Härtel.

[18] Susan Foster. 2010. *Choreographing Empathy: Kinesthesia in Performance*. Routledge.

[19] E. Bruce Goldstein and James Brockmole. 2016. *Sensation and Perception*. Cengage Learning.

[20] Maas Goudswaard, Abel Abraham, Bruna Goveia da Rocha, Kristina Andersen, and Rong-Hao Liang. 2020. FabriClick: Interweaving pushbuttons into fabrics using 3D printing and digital embroidery. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. 379–393.

[21] Tamil Selvan Gunasekaran, Ryo Hajika, Chloe Dolma Si Ying Haigh, Yun Suen Pai, Danielle Lottridge, and Mark Billinghurst. 2021. Adapting fitts' law and n-back to assess hand proprioception. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[22] Tamil Selvan Gunasekaran, Ryo Hajika, Yun Suen Pai, Eiji Hayashi, and Mark Billinghurst. 2022. RaITIn: Radar-based identification for tangible interactions. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA,) *(CHI EA'22)*. ACM, New York, , Article 445, 7 pages. https://doi.org/10.1145/3491101.3519808

[23] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 283–292.

[24] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. 2013. Understanding palm-based imaginary interfaces: The role of visual and tactile cues when browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 889–898.

[25] Jia Han, Gordon Waddington, Roger Adams, Judith Anson, and Yu Liu. 2016. Assessing proprioception: A critical review of methods. *Journal of Sport and Health Science* 5, 1 (2016), 80–90.

[26] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 441–450.

[27] Chris Harrison and Scott E. Hudson. 2010. Minput: Enabling interaction on small mobile devices with high-precision, low-cost, multipoint optical tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1661–1664.

[28] 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*. 453–462.

[29] Eiji Hayashi, Jaime Lien, Nicholas Gillian, Leonardo Giusti, Dave Weber, Jin Yamanaka, Lauren Bedal, and Ivan Poupyrev. 2021. RadarNet: Efficient gesture recognition technique utilizing a miniature radar sensor. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[30] Susan Hillier, Maarten Immink, and Dominic Thewlis. 2015. Assessing proprioception: A systematic review of possibilities. *Neurorehabilitation and Neural Repair* 29, 10 (2015), 933–949.

[31] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Adam Hartwig. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[32] Roland S. Johansson and A. B. Vallbo. 1979. Tactile sensibility in the human hand: Relative and absolute densities of four types of mechanoreceptive units in glabrous skin. *The Journal of Physiology* 286, 1 (1979), 283–300.

[33] Chang-Yong Kim, Jong-Duk Choi, and Hyeong-Dong Kim. 2014. No correlation between joint position sense and force sense for measuring ankle proprioception in subjects with healthy and functional ankle instability. *Clinical Biomechanics* 29, 9 (2014), 977–983.

[34] Dae-Hyeong Kim, Nanshu Lu, Rui Ma, Yun-Soung Kim, Rak-Hwan Kim, Shuodao Wang, Jian Wu, Sang Min Won, Hu Tao, Ahmad Islam, et al. 2011. Epidermal electronics. *Science* 333, 6044 (2011), 838–843.

[35] Gierad Laput, Robert Xiao, Xiang'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin buttons: Cheap, small, low-powered and clickable fixed-icon laser projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 389–394.

[36] 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)*. ACM, New York, , 321–333. https://doi.org/10.1145/2984511.2984582

[37] Rong-Hao Liang, Shu-Yang Lin, Chao-Huai Su, Kai-Yin Cheng, Bing-Yu Chen, and De-Nian Yang. 2011. SonarWatch: Appropriating the forearm as a slider bar. In *Proceedings of SIGGRAPH Asia 2011 Emerging Technologies*. 1–1.

[38] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–19.

[39] Shu-Yang Lin, Chao-Huai Su, Kai-Yin Cheng, Rong-Hao Liang, Tzu-Hao Kuo, and Bing-Yu Chen. 2011. Pub-point upon body: Exploring eyes-free interaction and methods on an arm. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 481–488.

[40] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 63–71.

[41] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. 2020. Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[42] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. 2020. Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1, Article 20 (March 2020), 27 pages. https://doi.org/10.1145/3380984

[43] Joseph A. Maldjian, Allan Gottschalk, Rita S. Patel, John A. Detre, and David C. Alsop. 1999. The sensory somatotopic map of the human hand demonstrated at 4 Tesla. *Neuroimage* 10, 1 (1999), 55–62.

[44] Flavia Mancini, Armando Bauleo, Jonathan Cole, Fausta Lui, Carlo A. Porro, Patrick Haggard, and Gian Domenico Iannetti. 2014. Whole-body mapping of spatial acuity for pain and touch. *Annals of Neurology* 75, 6 (2014), 917–924.

[45] Atsuo Murata and Hirokazu Iwase. 2001. Extending Fitts' law to a three-dimensional pointing task. *Human Movement Science* 20, 6 (2001), 791 – 805.

[46] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 177–186.

[47] Aditya Shekhar Nittala, Arshad Khan, and Jürgen Steimle. 2020. Conformal wearable devices for expressive on-skin interaction. In *Proceedings of the Augmented Humans International Conference*. 1–3.

[48] Aditya Shekhar Nittala, Klaus Kruttwig, Jaeyeon Lee, Roland Bennewitz, Eduard Arzt, and Jürgen Steimle. 2019. Like a second skin: Understanding how epidermal devices affect human tactile perception. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–16.

[49] Masa Ogata and Michita Imai. 2015. SkinWatch: Skin gesture interaction for smart watch. In *Proceedings of the 6th Augmented Human International Conference*. 21–24.

[50] Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: Adapting skin as a soft interface. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. 539–544.

[51] Michael Otero. 2005. Application of a continuous wave radar for human gait recognition. In *Signal Processing, Sensor Fusion, and Target Recognition XIV*, Vol. 5809. International Society for Optics and Photonics, 538–548.

[52] Siyou Pei, Alexander Chen, Jaewook Lee, and Yang Zhang. 2022. Hand interfaces: Using hands to imitate objects in AR/VR for expressive interactions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI'22)*. ACM, New York, , Article 429, 16 pages. https://doi.org/10.1145/3491102.3501898

[53] Valeria Peviani and Gabriella Bottini. 2020. Proprioceptive errors in the localization of hand landmarks: What can be learnt about the hand metric representation? *PLOS One* 15, 7 (2020), e0236416.

[54] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-defined gestures for augmented reality. In *Proceedings of the IFIP Conference on Human-Computer Interaction*. Springer, 282–299.

[55] Uwe Proske and Simon C. Gandevia. 2012. The proprioceptive senses: Their roles in signaling body shape, body position and movement, and muscle force. *Physiological Reviews* 92, 4 (2012), 1651–1697.

[56] Tauhidur Rahman, Alexander T. Adams, Ruth Vinisha Ravichandran, Mi Zhang, Shwetak N. Patel, Julie A. Kientz, and Tanzeem Choudhury. 2015. Dopplesleep: A contactless unobtrusive sleep sensing system using short-range doppler radar. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 39–50.

[57] Reza Rawassizadeh, Blaine A. Price, and Marian Petre. 2014. Wearables: Has the age of smartwatches finally arrived? *Commun. ACM* 58, 1 (dec 2014), 45–47. https://doi.org/10.1145/2629633

[58] Julius Cosmo Romeo Rudolph, David Holman, Bruno De Araujo, Ricardo Jota, Daniel Wigdor, and Valkyrie Savage. 2022. Sensing hand interactions with everyday objects by profiling wrist topography. In *Proceedings of the 16th*

International Conference on Tangible, Embedded, and Embodied Interaction (Daejeon, Republic of Korea) *(TEI'22)*. ACM, New York, Article 14, 14 pages. https://doi.org/10.1145/3490149.3501320

[59] Avik Santra, Raghavendran Vagarappan Ulaganathan, Thomas Finke, Ashutosh Baheti, Dennis Noppeney, Jungmaier Reinhard Wolfgang, and Saverio Trotta. 2018. Short-range multi-mode continuous-wave radar for vital sign measurement and imaging. In *Proceedings of the 2018 IEEE Radar Conference (RadarConf18)*. IEEE, 0946–0950.

[60] Jacob L. Segil, Ivana Cuberovic, Emily L. Graczyk, Dustin Tyler, et al. 2020. Combination of simultaneous artificial sensory percepts to identify prosthetic hand postures: A case study. *Scientific Reports* 10, 1 (2020), 1–15.

[61] Teddy Seyed, Chris Burns, Mario Costa Sousa, Frank Maurer, and Anthony Tang. 2012. Eliciting usable gestures for multi-display environments. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*. 41–50.

[62] Charles Sherrington. 1952. *The Integrative Action of the Nervous System*. CUP Archive.

[63] Myrim Sillevis Smitt and H. A. Bird. 2013. Measuring and enhancing proprioception in musicians and dancers. *Clinical Rheumatology* 32, 4 (2013), 469–473.

[64] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and above-skin input sensing through a wearable depth sensor. In *Proceedings of ACM CHI*. 12 pages. http://handtracker.mpi-inf.mpg.de/projects/WatchSense/

[65] Jürgen Steimle, Joanna Bergstrom-Lehtovirta, Martin Weigel, Aditya Shekhar Nittala, Sebastian Boring, Alex Olwal, and Kasper Hornbæk. 2017. On-skin interaction using body landmarks. *Computer* 50, 10 (2017), 19–27.

[66] Barry C. Stillman. 2002. Making sense of proprioception: The meaning of proprioception, kinaesthesia and related terms. *Physiotherapy* 88, 11 (2002), 667–676.

[67] 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,) *(CHI'22)*. ACM, New York, , Article 497, 16 pages. https://doi.org/10.1145/3491102.3501878

[68] Yuta Sugiura, Fumihiko Nakamura, Wataru Kawai, Takashi Kikuchi, and Maki Sugimoto. 2017. Behind the palm: Hand gesture recognition through measuring skin deformation on back of hand by using optical sensors. In *Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. IEEE, 1082–1087.

[69] William Taube Navaraj, Carlos García Núñez, Dhayalan Shakthivel, Vincenzo Vinciguerra, Fabrice Labeau, Duncan H. Gregory, and Ravinder Dahiya. 2017. Nanowire FET based neural element for robotic tactile sensing skin. *Frontiers in Neuroscience* 11 (2017), 501.

[70] John C. Tuthill and Eiman Azim. 2018. Proprioception. *Current Biology* 28, 5 (2018), R194-R203. https://doi.org/10.1016/j.cub.2018.01.064

[71] Radu-Daniel Vatavu. 2012. User-defined gestures for free-hand TV control. In *Proceedings of the 10th European Conference on Interactive TV and Video*. 45–48.

[72] Klen Čopič Pucihar, Nuwan T. Attygalle, Matjaz Kljun, Christian Sandor, and Luis A. Leiva. 2022. Solids on soli: Millimetre-wave radar sensing through materials. *Proc. ACM Hum.-Comput. Interact.* 6, EICS, Article 156 (Jun 2022), 19 pages. https://doi.org/10.1145/3532212

[73] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 851–860.

[74] Yazhou Wang and Aly E. Fathy. 2011. Micro-Doppler signatures for intelligent human gait recognition using a UWB impulse radar. In *Proceedings of the 2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, 2103–2106.

[75] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. 2005. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*. 99–104.

[76] Robert Watson-Watt. 1945. Radar in war and in peace.

[77] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. Iskin: Flexible, stretchable and visually customizable on-body touch sensors for mobile computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. https://doi.org/10.1145/2702123.27023911315

[78] Martin Weigel, Vikram Mehta, and Jürgen Steimle. 2014. More than touch: Understanding how people use skin as an input surface for mobile computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI'14)*. Association for Computing Machinery, New York, NY, USA, 179–188. https://doi.org/10.1145/2556288.2557239

[79] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. 2017. Skinmarks: Enabling interactions on body landmarks using conformal skin electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3095–3105.

[80] Anusha Withana, Daniel Groeger, and Jürgen Steimle. 2018. Tacttoo: A thin and feel-through tattoo for on-skin tactile output. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 365–378.

[81] 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*. 1147–1160.

[82] A. S. Wycherley, P. S. Helliwell, and H. A. Bird. 2005. A novel device for the measurement of proprioception in the hand. *Rheumatology* 44, 5 (2005), 638–641.

[83] Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. 2018. LumiWatch: On-arm projected graphics and touch input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[84] 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,) *(CHI'22)*. ACM, New York, Article 496, 19 pages. https://doi.org/10.1145/3491102.3501904

[85] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. 2018. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[86] Hui-Shyong Yeo, Gergely Flamich, Patrick Schrempf, David Harris-Birtill, and Aaron Quigley. 2016. Radarcat: Radar categorization for input & interaction. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 833–841.

[87] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T. Inan, Thad E. Starner, and Gregory D. Abowd. 2016. TapSkin: Recognizing on-skin input for smartwatches. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (Niagara Falls, Ontario, Canada) *(ISS'16)*. ACM, New York, 13–22. https://doi.org/10.1145/2992154.2992187

[88] Maotian Zhang, Qian Dai, Panlong Yang, Jie Xiong, Chang Tian, and Chaocan Xiang. 2018. IDial: Enabling a virtual dial plate on the hand back for around-device interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 55 (mar 2018), 20 pages. https://doi.org/10.1145/3191787

[89] 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*. 1491–1503.

[90] Yan Zhuang, Chen Song, Aosen Wang, Feng Lin, Yiran Li, Changzhan Gu, Changzhi Li, and Wenyao Xu. 2015. SleepSense: Non-invasive sleep event recognition using an electromagnetic probe. In *Proceedings of the 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 1–6.

[91] Sasha N. Zill. 2019. Mechanoreceptors: Exteroceptors and proprioceptors. In *Cockroaches as Models for Neurobiology: Applications in Biomedical Research*. CRC Press, 247–267.

Received 8 July 2022; revised 20 June 2023; accepted 23 June 2023