Design and Recognition of Microgestures for Always-Available Input

UNIVERSITÄT
DES
SAARLANDES

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Adwait Sharma

Saarbrücken, 2022

Day of colloquium:	21st July 2022
Dean of the Faculty:	Prof. Dr. Jürgen Steimle
Reviewers:	Prof. Dr. Jürgen Steimle (Saarland University, Germany) Prof. Dr. Daniel Vogel (University of Waterloo, Canada)
Chair of the committee:	Prof. Dr. Antonio Krüger (Saarland University and DFKI, Germany)
Academic assistant:	Dr. Alice Haynes (Saarland University, Germany)

to my inspiring teachers and loving family...

Abstract

Gestural user interfaces for computing devices most commonly require the user to have at least one hand free to interact with the device, for example, moving a mouse, touching a screen, or performing mid-air gestures. Consequently, users find it difficult to operate computing devices while holding or manipulating everyday objects. This limits the users from interacting with the digital world during a significant portion of their everyday activities, such as, using tools in the kitchen or workshop, carrying items, or workout with sports equipment.

This thesis pushes the boundaries towards the bigger goal of enabling always-available input. Microgestures have been recognized for their potential to facilitate direct and subtle interactions. However, it remains an open question how to interact using gestures with computing devices when both of the user's hands are occupied holding everyday objects. We take a holistic approach and focus on three core contributions: i) To understand end-users preferences, we present an empirical analysis of users' choice of microgestures when holding objects of diverse geometries. Instead of designing a gesture set for a specific object or geometry and to identify gestures that generalize, this thesis leverages the taxonomy of grasp types established from prior research. ii) We tackle the critical problem of avoiding false activation by introducing a novel gestural input concept that leverages a single-finger movement, which stands out from everyday finger motions during holding and manipulating objects. Through a data-driven approach, we also systematically validate the concept's robustness with different everyday actions. iii) While full sensor coverage on the user's hand would allow detailed hand-object interaction, minimal instrumentation is desirable for real-world use. This thesis addresses the problem of identifying sparse sensor layouts. We present the first rapid computational method, along with a GUI-based design tool that enables iterative design based on the designer's high-level requirements. Furthermore, we demonstrate that minimal form-factor devices, like smart rings, can be used to effectively detect microgestures in hands-free and busy scenarios.

Overall, the presented findings will serve as both conceptual and technical foundations for enabling interaction with computing devices wherever and whenever users need them.

Zusammenfassung

Benutzerschnittstellen für Computergeräte auf Basis von Gesten erfordern für eine Interaktion meist mindestens eine freie Hand, z.B. um eine Maus zu bewegen, einen Bildschirm zu berühren oder Gesten in der Luft auszuführen. Daher ist es für Nutzer schwierig, Geräte zu bedienen, während sie Gegenstände halten oder manipulieren. Dies schränkt die Interaktion mit der digitalen Welt während eines Großteils ihrer alltäglichen Aktivitäten ein, etwa wenn sie Küchengeräte oder Werkzeug verwenden, Gegenstände tragen oder mit Sportgeräten trainieren.

Diese Arbeit erforscht neue Wege in Richtung des größeren Ziels, immer verfügbare Eingaben zu ermöglichen. Das Potential von Mikrogesten für die Erleichterung von direkten und feinen Interaktionen wurde bereits erkannt. Die Frage, wie der Nutzer mit Geräten interagiert, wenn beide Hände mit dem Halten von Gegenständen belegt sind, bleibt jedoch offen. Wir verfolgen einen ganzheitlichen Ansatz und konzentrieren uns auf drei Kernbeiträge: i) Um die Präferenzen der Endnutzer zu verstehen, präsentieren wir eine empirische Analyse der Wahl von Mikrogesten beim Halten von Objekte mit diversen Geometrien. Anstatt einen Satz an Gesten für ein bestimmtes Objekt oder eine bestimmte Geometrie zu entwerfen, nutzt diese Arbeit die aus früheren Forschungen stammenden Taxonomien an Griff-Typen. ii) Wir adressieren das Problem falscher Aktivierungen durch ein neuartiges Eingabekonzept, das die sich von alltäglichen Fingerbewegungen abhebende Bewegung eines einzelnen Fingers nutzt. Durch einen datengesteuerten Ansatz validieren wir zudem systematisch die Robustheit des Konzepts bei diversen alltäglichen Aktionen. iii) Auch wenn eine vollständige Sensorabdeckung an der Hand des Nutzers eine detaillierte Hand-Objekt-Interaktion ermöglichen würde, ist eine minimale Ausstattung für den Einsatz in der realen Welt wünschenswert. Diese Arbeit befasst sich mit der Identifizierung reduzierter Sensoranordnungen. Wir präsentieren die erste, schnelle Berechnungsmethode in einem GUI-basierten Designtool, das iteratives Design basierend auf den Anforderungen des Designers ermöglicht. Wir zeigen zudem, dass Geräte mit minimalem Formfaktor wie smarte Ringe für die Erkennung von Mikrogesten verwendet werden können.

Insgesamt dienen die vorgestellten Ergebnisse sowohl als konzeptionelle als auch als technische Grundlage für die Realisierung von Interaktion mit Computergeräten wo und wann immer Nutzer sie benötigen.

Acknowledgements

I will always be indebted to all the outstanding people who have helped me grow professionally and personally. I feel fortunate to be encouraged and influenced by them.

Firstly, I want to thank my advisor, Prof. Jürgen Steimle, whose wisdom and kindness have profoundly influenced me. I cannot thank him enough for running a lab where everyone is encouraged to pursue their interests. I still vividly remember my days as an intern during my master's in 2016, and since then, you have always been there to help with everything I needed. Thanks to the right amount of intellectual challenges and encouragement you have provided me, I have become a better researcher and person.

I'd like to thank Prof. Daniel Ashbrook, my unofficial advisor. The opportunity to learn directly from someone whose seminal work on microgestures I admire was an incredible privilege. I am grateful that you took the time to engage in our weekly discussions and made me feel welcome in Copenhagen.

I'd also like to thank Prof. Thomas Seel for being so generous with the hardware used in IMU experiments and for going above and beyond in helping me conduct the study, particularly during the peak times of COVID-19 pandemic.

I owe special thanks to Prof. Daniel Vogel for being on my committee. Even before I began my PhD, your work on interaction techniques was an inspiration to me. Perhaps this is a good time to admit that I always look for your name in the conference proceedings.

I am also grateful to the postdocs with whom I co-authored papers: Joan Sol Roo, Bruno Fruchard, Jess McIntosh, Christina Salchow-Hömmen, and Marion Koelle, who not only spent countless hours discussing even the tiniest project details but also shared their personal experiences and patiently worked with me.

I am thankful to the Bosch Research's Human-Machine Interface team, especially Thorsten Sohnke and Frank Beruscha, for fruitful discussions regarding practical use cases of my research. In addition, I improved my presentation skills by sharing my findings and through showcasing application demos. Bosch Research and the European Research Consortium (ERC) grant funded my work in this thesis; I thank Jürgen for facilitating these unique opportunities and funding me.

A thank you will not suffice to express my gratitude to my excellent advisors from my early career and during my master's studies: Prof. Ellen Yi-Luen Do and Prof. Michael Haller. In a humbly nurturing manner, they exposed me to the world of Human-Computer Interaction and cultivated my scientific curiosity, which led me to pursue this career.

Special thanks to Aditya Shekhar Nittala - my close friend, officemate, co-author, and someone who sets high standards in everything he does. Your help has been vital to my success, from the lame puns we cracked in the office to your readiness to help me solve complex research problems.

Another special thanks to Michael A. Heddrich, with whom I started hanging out at the beginning of my PhD. I had no idea we would become such close friends that we would explore almost every vegetarian restaurant in town, play bagpipe, go on a road trip, and co-author papers. I have learned so many things from you, both personally and professionally. Also, my sincere thanks to his advisor, Prof. Dietrich Klakow, for providing his invaluable advice on the project.

Mentoring some extremely talented students for their thesis or during their internship has greatly expanded my horizons. A big thank you to: Alejandro Helgueros, Divyanshu Bhardwaj, Muhammad Habib, Luca Bläsius, and Vimal Suresh Mollyn.

There is a certain joy in coming to lab every morning because my labmates, past and present, make it a rewarding place to work: The *intern of the month* award from Daniel Gröger and the coffee conversations with Martin Weigel made me feel welcome in the group from my internship days; Even though I never got the chance to collaborate closely with Anusha Withana, Marc Teyssier, Arshad Khan, and Paul Strohmeier, I thank them for sharing their stories to encourage me; Kind thanks to Manoz Banda for treating me with the spicy food; Thanks to Narges Pourjafarian for helping me with my long list of moodle-related questions; Thanks to Alice Haynes for proofreading parts of this thesis and bearing the US english text; Thanks to Marie Mühlhaus as well for always helping me with design-related queries and translating the German version of this thesis abstract; Thanks, Artin Saberpour, for all the fun conversations in the corridor; Thank you Ata Otaran and Madalina Nicolae, for reminding me to take breaks and joining me for ice cream.

Thanks to the administrative support team, including Anne Bardesano-Kopsel, Asel Bojarski, Evelyn Kraska, Kathrin Keim, Mona Linn, Natalia Weise, and Polina Quaranta, who helped me to simplify the paperwork and regulations.

Special thanks to Shievam Kashyap, who has always cheered me up and is always there to help me. It's always fascinating to learn about so many different topics from you. I am deeply thankful for our friendship.

Lastly, I cannot express in words the support and affection I have received from my parents and my sister, as well as their sacrifices to make me the person I am today.

Contents

CHAPTER 1

Introduction

> "The hand is the visible part of the brain."
>
> *Immanuel Kant*

Computers are no longer limited to desktops, workstations, and laptops; they have shrunk from room-sized machines to handheld devices such as smartphones, tablets, and smartwatches. Despite the integration of other interesting input modalities like eye-gaze and voice into new devices, gestural input with hands remains the most prevalent, enabling both precise *continuous* and quick *discrete* gestures. Although the trend toward smaller form factors has created devices that can be carried in users' pockets or worn on the wrist, users still need at least one hand free to interact with the devices, for instance, to be able to touch a screen, hold a controller or perform mid-air gestures.

However, human hands are constantly busy in myriad contexts of everyday life, e.g., holding utensils while cooking or eating at home, writing with a pen at the office, carrying shopping bags down the street, and using a hammer or other repairing tools in the workshop. Thus, users are unable to interact with their devices while they are performing these activities, which is a challenging problem that deprives them of always-available input. In particular, the diverse geometry of objects present in these activities and the interactional constraints caused by holding a physical object in one's hand add complexity to gesture design and recognition. Introducing an input technique in such everyday situations will enable varied real-world applications — ranging from skipping a music track on-the-go to controlling critical healthcare systems.

In this thesis, I holistically approach this problem while balancing the needs of the task at hand and human capabilities. The approach includes drawing insights from users' behavior, and capturing and performing quantitative analyses of large datasets. Furthermore, leveraging machine learning techniques to address the complexity and develop applications that effectively enable always-available input.

1.1 Research Challenges

In terms of usability, a practical solution should have consistent gestures across different objects, which a minimally invasive and low-cost sensing technique can detect. Toward satisfying these requirements and designing an end-to-end system, this thesis takes a systematic approach by first understanding the relationship between all five fingers and various grasp types, then introducing a novel input technique that is resilient to false activations, and finally providing a computational design tool for rapidly selecting a sparse sensor layout. The tool enables effective gesture recognition in both hands-free and busy scenarios with minimal hand instrumentation. Specifically, in my effort toward the vision of providing always-available input, I aim to answer the following three research questions:

RQ I: How does the multitude of grasp types and object geometries affect users' choice of microgestures?

Digital input methods that include microgestures have already proven to be useful for direct and subtle interaction with ubiquitous computing systems [1–3]. Prior work has systematically investigated single-hand microgestures in hands-free context [4]. However, gestures made in hands-free scenarios are likely to differ considerably from gestures performed when the hands are occupied holding an object. The number of fingers needed for holding or manipulating a handheld object largely constrains the set of possible microgestures, a scenario that is comparatively under-investigated in the literature. Pioneering work by Wolf et al. [5] has contributed an early investigation with 3 objects, while other work has investigated gestures on self-sustained objects, such as the steering wheel [6]. However, we still lack a systematic investigation of a more comprehensive set of object geometries and their respective grasps to investigate the complex relationship between handheld objects and microgestures. It remains an open question as to what are appropriate gestures from an end user's perspective when hands are busy holding an object.

RQ II: How to avoid false activations in gestural input while handling everyday objects?

While performing quick and convenient gestures on handheld objects is compelling as an input modality, gestures risk conflicting with finger movements that might occur when adjusting ones grip or manipulating the object. As a result, a gesture recognizer might misinterpret natural finger movements as an intentional input gesture and trigger an unintended command; namely, a *false activation.* This is a challenging problem for designing and deploying microgestural interfaces. Yet, to the best of our knowledge, there is no prior work that systematically investigates robust gestures to perform while holding everyday objects. Previous work has presented robust gestures for specific devices, such as smartphones [7–9], tablets [10], and smart pens [11]. While these approaches perform well in their specific contexts, a critical limitation is their device-specific behavior. For example, a flipping gesture might be suitable for smartphones [7], but would not work while holding a coffee cup. To address this, other work has proposed using a robust delimiter gesture that must be performed before doing the actual gesture [12]. An alternative approach could be to involve complicated movements that are unlikely to happen in everyday actions, such as a specific movement sequences, specific timing, or specific finger combinations. As opposed to this, it is desirable to incorporate input techniques that are not device-specific but ensure compatibility with a wide range of grasps and daily actions. Also, the technique should avoid the cognitive overhead associated with separate delimiters or complicated-to-perform gestures.

RQ III: What sensor locations on the hand provide effective recognition with minimal instrumentation?

Implementing a system to detect microgestures in both free-hand and busy-hand conditions poses significant technical challenges. Apart from the numerous sensor placement configurations that can be used to effectively detect dexterous finger movements, recognition systems need to deal with visual occlusions arising when the hands are occupied. These challenges make the deployment of optical sensing techniques very demanding [13]. A common approach to address hand occlusion challenges when manipulating objects is to use data gloves or a large number of markers. For instance, Han et al. [14] achieved promising results for hand pose reconstruction while manipulating objects by employing deep learning combined with markers attached all over the hand. Yet, extensive hand instrumentation is undesirable for practical use because it would be cumbersome and interfere with the user's daily routine (e.g., handling food items or utensils in the kitchen). The sparse sensing principles provide a technically efficient solution by placing sensors in an optimal location. Additionally, it could be used to detect microgestures in real-world scenarios through a minimal form factor device like smart ring(s).

1.2 Summary of Contributions

To approach the problem of providing always-available gestural input with handheld objects, this thesis investigates various aspects of gestural interface design, from iterative design to technical implementation. Moreover, as a first step in developing a comprehensive understanding and providing input with diverse object geometries, I investigated all six prehensile postures described in Schlesinger's seminal grasp taxonomy [15] that incorporates object shape, hand surface, and hand shape. The taxonomy is widely used in prior work across domains [12, 16–19]. Specifically, this thesis includes a formative **empirical** study about the relationship between finger movements and grasps, **design and validation** of a novel gestural concept to avoid false activations, and a **method** to utilize a minimal amount of sensors for live gesture recognition. Below is a brief description of contributions to each research question:

I: Consolidated gesture set while grasping diverse everyday objects

Given the wide variety of object geometries that we hold or use during our everyday lives, the first aim was to investigate how the multitude of grasps and object geometries affect users' choice of microgestures. For instance, would application designers of mobile computing and the Internet-of-Things have to design a custom gesture set for each object? From a usability standpoint, it would be highly undesirable and would risk frustrating users up to the point of rejecting the new opportunities unleashed by such microgestures. To answer these questions, I used the elicitation method in an experiment with end-users holding a variety of objects, from as small as a needle to a large box. Subsequently, the design recommendations are derived for *Grasping Microgestures* [Sharma et al. 20], a class of gestures performed by the same hand involved in holding the object. As opposed to most approaches that have designed gestures based on a particular object/geometry (e.g., pen), I focused on the generalisability and scalability of the gestures by taking into account a complete taxonomy of grasp types established from prior work. A full data-driven interpretation using statistical clustering revealed previously undiscovered patterns. It identified similarities among different grasp types and ultimately allowed us to present three main cluster sets of gestures that cover interactions across varied grasp types. Furthermore, these clusters inform how grasps and object geometries affect single-hand microgestures, preferred locations, and fingers used.

II: Avoid false activations during everyday hand-object actions

Grasping Microgestures demonstrated the possibility of performing gestures with different grasp types. However, there is a risk of confusing user-defined gestures with an object or task-related finger motions. Previous work presented individual device-specific delimiter gestures that would not be compatible with everyday objects. Also, such delimiting gestures create an interruption of the task at hand. In contrast, I concentrated on gestures' compatibility with a wide range of grasps and everyday actions. This thesis introduces the concept of *SoloFinger* [Sharma et al. 21] – leveraging the insight that fingers tend to be static, or multiple fingers simultaneously move when holding and manipulating objects. Consequently, moving a single finger to a considerable yet comfortable extent while all other fingers remain static stands out. This idea opens a space for creating robust yet easy-to-perform gestures. It also allows for versatile gesture variations (e.g., moving a finger forward, backward, or drawing a pattern like circle or zigzag). A series of data-driven analyses was performed on a pre-existing dataset with 36 everyday hand-object actions to validate the SoloFinger concept. Specifically, the analysis includes formulating a new metric (PeakScore) to quantify the single-finger movement and developing a white-box classification technique for human interpretability of the classification errors. In addition, I also demonstrated the concept's real-world feasibility by implementing an end-to-end system with a commercially available virtual reality glove. When the held object is known, the multi-class gesture classification accuracy was 89%, without any false activation in the collected dataset.

III: Sparse sensor placement for sensing fine-grained finger microgestures in freehand and grasping conditions

Capturing finger gestures with optical-based sensing is prone to occlusion when hands are busy holding or carrying objects. Although glove-based solutions mitigate this issue, they impede usability because of their bulkiness. Inertial measurement units (IMUs) provide a better solution to effectively recognize microgestures in freehand contexts and while grasping everyday objects. However, their count and placement are crucial for effective gesture recognition. Minimal form factor devices can be designed using the principles of *sparse sensing* (also known as compressed sensing). For example, two smart rings worn on optimally selected finger segments can overcome the aforementioned technical limitations. Designing an IMU layout that is sparse is a difficult task due to the complex multi-factorial space, which includes freehand or grasping conditions, diverse object geometries, different fingers, a multitude of gestures, and additional user-defined constraints. Considering the complexity of the multi-factorial design space, this manual process is time-consuming and may lead to far sub-optimal layouts. I tackled this problem by creating a computational design approach, which started with collecting the Microgestures dataset. In particular, through customized hardware with 17 synchronized IMUs, I captured microgestures and hand manipulations when users were Freehand and Grasping 12 objects. Consequently, this thesis introduces the first rapid computational method, *SparseIMU* [Sharma et al. 22] to generate sparse layouts for detecting microgestures. I employ a variant of a well-known metric from Machine Learning (ML), Feature Importance, to rapidly select sparse layouts. Compared to the combinatorial search that takes hours or days on a 40-core cluster system, the SparseIMU method can generate the layout within a few seconds or minutes on a commodity laptop. Notably, the F1 Score from the layout generated by the SparseIMU method is similar to the best layouts found by training all 400K layout combinations.

This thesis also presents a GUI-based design tool to help the designers/engineers rapidly find optimal configurations for IMU-based sensing devices (such as smart rings or other wearable devices) with minimal hand instrumentation. Furthermore, in-depth analysis to quantify performance across different possible IMU layouts uncovered the potential of IMU sensing to detect microgestures.

1.3 Publications

I have published full papers based on the ideas presented in this dissertation at the ACM Conference on Human Factors in Computing Systems (CHI) [P1, P2], and a journal article at ACM Transactions on Computer-Human Interaction (TOCHI) [P3]:

P1. Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. "Grasping Microgestures: Eliciting Single-hand Microgestures for Handheld Objects." In: *Proceedings of the 37th Annual ACM Conference on Human Factors in Computing Systems (CHI '19)*, 13 pages [20].

P2. Adwait Sharma, Michael A. Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. "SoloFinger: Robust Microgestures while Grasping Everyday Objects." In: *Proceedings of the 39th Annual ACM Conference on Human Factors in Computing Systems (CHI '21)*, 15 pages [21].

P3. Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. "SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-Grained Finger Microgestures." In: *ACM Transactions on Computer-Human Interaction (TOCHI '22)*, 40 pages [22].

1.4 Thesis Roadmap

This thesis is organized into six chapters. The next chapter presents the related literature, including taxonomies of grasping, design of freehand and grasping microgestures, and false activation during gestural input. It additionally covers the state-of-the-art techniques of gesture sensing, recognition, and computational design tools. Chapters 3 investigates the user-defined microgestures across different grasp types. Chapter 4 introduces SoloFinger, a novel concept to avoid false activations. Chapter 5 describes a computational method and a GUI-based tool for designing sparse sensor layouts. Finally, Chapter 6 provides the conclusion and opportunities for several future research directions.

CHAPTER 2

Related Work

Designing and implementing always-available input that caters to a wide group of users requires expertise and scientific endeavors from multiple research disciplines, including human-computer interaction, machine learning, and optimization. This chapter focuses on the related work about crucial areas, from understanding prehensile hand poses while grasping diverse object geometries, avoiding false activations during input, to sensing and recognizing gestures efficiently.

2.1 Grasp taxonomies

We use our hands for a variety of daily activities, and many occupations even depend entirely on them, including but not limited to the chef, warehouse worker, surgeon, and mechanic. Here, it is worth noting that our hands are constantly busy grasping and manipulating diverse objects to perform the task at hand, e.g., reading a book or carrying grocery bags. Heo et al. [23] and Laput et al. [24] compiled a list of such hands-busy scenarios by reviewing prior literature and conducting an in-the-wild study, respectively.

Previous research has categorized hand-object interaction in three states: *Off-hand* (no physical interaction with the object), *In contact* (hand approaching the object), and *Held in-hand* (hand grasping the object) [25]. Considering the scope of this thesis, the state of Held in-hand is more relevant. Further, grasp taxonomies are used to categorize various hand poses necessary to hold different object geometries. Several taxonomies of discrete grasp have been proposed for various goals. Notably, Schlesinger [15] put forth a seminal taxonomy initially developed by considering the functionality required for the prosthetic hands. Napier in 1956 [26] presented two basic grips, namely precision and power, that are derived from anatomical and functional views. Cutkosky [27] derived a more detailed classification of grasp types by conducting a study in a manufacturing operation. Feix et al. [28] analyzed the literature and presented a taxonomy with 33 different grasp types. This taxonomy encompasses several grasp types, but it's important to note that for only a subset of object geometries, it distinguishes between different object sizes and separates grasps based on which finger is grasping the same object. In our work, we investigated all six prehensile postures from Schlesinger's taxonomy (illustrated in Figure 3.2). Additionally, we systematically included two sizes for each grasp type to accommodate a variety of objects. It is worth mentioning that this taxonomy has also been widely used in several previous studies [12, 16–19]. For a comprehensive survey of grasp taxonomies, we refer to MacKenzie and Iberall [29].

2.2 Microgesture design

HCI has a large body of work on gestural input techniques for interacting with the digital world, including expansive motion gestures based on using full-body [30–33] or arm movements [34–36]. Compared to these input methods that require significant physical movements, microgestures (or microinteractions) are performed through subtle movements that are fast, easy, and do not interrupt other ongoing tasks [37]. As a result, they are useful in various scenarios, including eyes-free use of mobile devices [1], sports activities [38], driving [6], and virtual reality [39]. Below we describe the concrete designs of microgestures using finger movement when hands are free and busy. Also, we will discuss strategies to avoid false inputs from the gesture design perspective.

2.2.1 Freehand and grasping

Previous work has identified the importance of including end-users in the gesture design process [40–44]. Morris et al. [42] have shown that users prefer gestures defined by larger groups over gestures created by a few researchers. The method of eliciting gestures from end-users, initially proposed by Wobbrock et al. [40], has quickly found widespread use in various areas, ranging from designing gestures for on-skin input [45] to interacting with drones [46]. Chan et al. [4] investigated properties of single-hand microgestures through an elicitation study in a freehand condition. The authors found that the end-users leveraged fingers' dexterity for designing a variety of gestures, and also a difference between all five fingers used for executing the gestures. These observations suggest that gestures and finger movements would further vary while holding an object.

However, little previous work has empirically investigated input while the user's hands are busy grasping an object. These mainly include self-sustained objects, such as steering wheels and bike handles [6, 47, 48] that allow the user to freely move their hand or fingers over the object without having the overhead to hold it. Our work is different in that users had to continuously hold the handheld objects. Lee et al. [49] explored deformation-based user gestures on various materials such as plastic, paper, and elastic cloth. We followed a

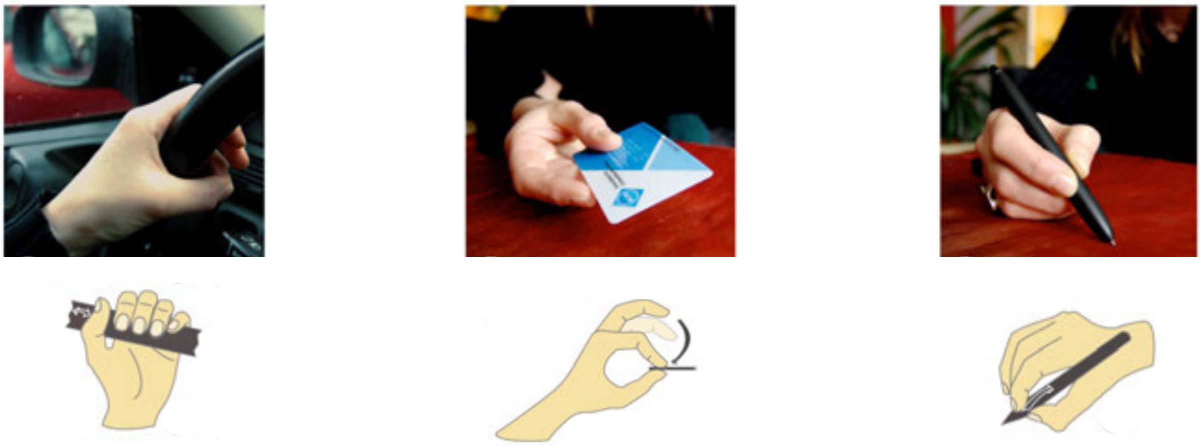


Figure 2.1: Microgestures designed while holding three objects: a steering wheel, cash card, and pen as presented by Wolf et al. [5]. (Image courtesy: Katrin Wolf).
.

similar approach using real-world objects. In our work, we leverage the gripping posture and embrace the challenge of using only one hand.

We drew inspiration from Fitzmaurice's seminal work published in 1995 with Graspable User Interfaces [50], which introduced the concept of grasping an object (physically or virtually) and leveraging its size, shape, and position for input in graphical user interfaces. Graspables [51] investigated input techniques by implementing physical form devices in the form of the soap bar and ball. Wolf et al. [5] investigated micro-interactions to support secondary tasks while the user's primary task involves holding an object by investigating three objects: steering wheel, cash card, and stylus as shown in Figure 2.1, wherein gestures are identified based on consultation with four experts. We extend this work by investigating a wider variety of 12 objects, conceptually based on Schlesinger's taxonomy of grasps (see Chapter 3). Based on a large set of gestures elicited from end-users, we contribute the first empirical analysis of how grasps and object size affect the properties of microgestures. In Chapter 5, we advance these conceptual foundations of user-defined unimanual gestures in freehand and grasping conditions through a sensing approach, facilitating real-world implementation.

2.2.2 Avoid false activations

Gesture detection errors can be classified in two categories: false positive errors, which relate to triggering unintended actions, and false negative errors, where a recognition system fails to identify the intended gesture. In HCI, both these errors result in user frustration and have direct implications on the adoption of a particular technology [52]. Since false negatives depend more on a particular system, we focus on a more general premise of avoiding false positives in our work.

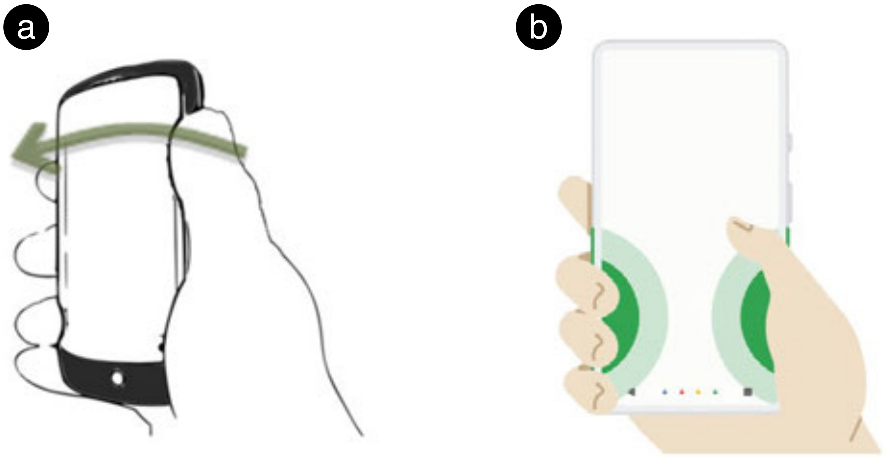


Figure 2.2: Unique gestures designed to reduce false activation on mobile phones. (a) DoubleFlip [7] enabled false activation reduction with flip gesture. (b) Active Edge [8] presented the idea of intentional squeeze as an intentional gesture. (Image courtesy: (a) Jaime Ruiz, (b) Philip Quinn)

Some previous work has focused on designing explicit *delimiter gestures* to avoid false activations. These gestures are significantly different from non-intentional actions and are therefore robust to unintended input. One common application for such delimiter gestures is switching between a gesture detection mode and another mode in which the gesture recognizer is not active. Through our literature review, we found two main categories of delimiter action for handheld objects: *bimanual gestures*, where the non-dominant hand performs the trigger action and, subsequently, the command is performed with the dominant hand [12]; and *device-specific* trigger actions like DoubleFlip [7] that require a large rotation, or Active Edge [8] that uses squeezing to detect intentional action on phone devices (see Figure 2.2). WristRotate [53] presented a wrist rotation technique as a delimiter for smartwatches. Recently, BlyncSync [54] used multi-modal touch and blink gestures on smartwatches. A drawback of any delimiter action is the disruption in the user's workflow: the user must first perform the delimiter, and then the intended gesture.

While one can create compound finger movements with Rhythmic microgestures [55], the technique demands practice and memorization. More closely related to our approach, Le et al. [56] explored reachability and unintended input for a specific grasp type with different phone sizes. From a technical standpoint, adding a large number of negative trials to a machine-learning algorithm reduces false positive rates [57, 58]. However, these techniques are applied after the design process of gestures. Secondly, they do not consider false positive reduction while performing the gestures during hand-object interactions. Other efforts include engaging end-users in the design process of gestures to reduce the risk of confusing gestures with natural movement [5, 20]. Our contribution is to employ a data-driven approach to validate a set of intentional gestures, which are resilient to false activation on un-instrumented everyday objects and various grasp types.

Our approach for designing gestures is inspired by Kawahata et al. [59], Magic 2.0 [60], and Gesture On [9] that compare gestures against a database to identify the ones most robust to false activations. In Chapter 4, we describe our more general approach with the design and validation of a gestural concept compatible with a broad range of grasp types, which leverages the unique single-finger motion with the same hand holding the object.

2.3 Microgesture sensing and recognition

Researchers have investigated different types of sensor data as well as several strategies to place them, including embedding sensors into the user's environment [61, 62], developing full touch-sensitive objects [63, 64], or as smart devices worn by the users [2, 65] to detect gestures (see Figure 2.3). Since wearing a device is more suitable for our goal of providing gestural input in diverse scenarios, we concentrated on this strategy.

2.3.1 Sensing type

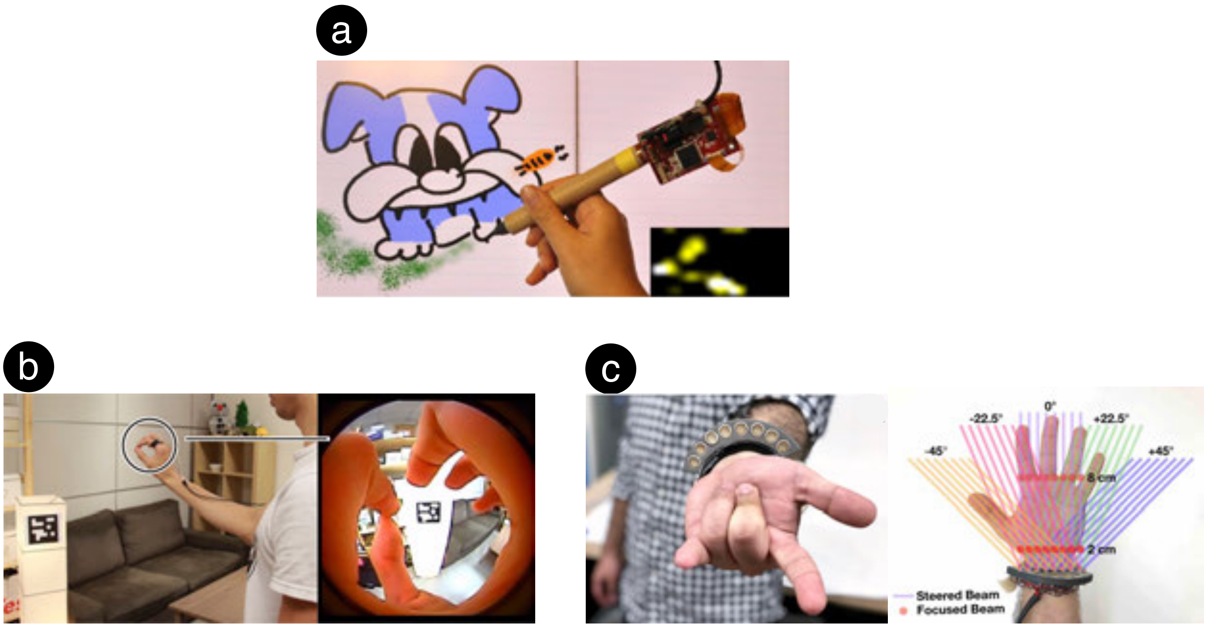


Figure 2.3: Exemplary approaches to sense hand gestures. (a) MTPen [63] developed a multi-touch pen by integrating a custom capacitive on the entire pen. (b) CyclopsRing [3] uses a fisheye camera to detect finger gestures in a ring-form factor device. (c) BeamBand [66] an array of ultrasonic transducers worn on the wrist. (Image courtesy: (a) Hyunyoung Song, (b) Liwei Chan, (c) Chris Harrison).

Various sensing techniques have been proposed to detect finger gestures. A large body of research relies on optical sensing for detecting microgestures. CyclopsRing [3] proposed a finger-worn fisheye camera device to detect on-finger and in-air pinch and slide gestures, as well as palm-writing, FingerInput [67] demonstrated detection of thumb-to-finger gestures using a head-mounted or shoulder-mounted depth sensor. Sugiura et al. [68] have shown recognition of discrete finger-based gestures using an array of photo reflective sensors placed on the back of hand. A variety of other sensing approaches include ultrasonic [66, 69, 70], infrared [2, 62, 69, 71], pressure [72–74], magnetic [75–77], and capacitive techniques [38, 78]. Due to the advances in deep learning, researchers have also demonstrated the detection of fine finger movements using radar sensing [61]. These systems show some remarkable success in enabling gesture recognition in freehand conditions. However, due to the inherent property of such sensing technologies, these approaches can fail under visual occlusion caused by holding an object. Attaching a sensor to an object can mitigate occlusion, but the this approach's scalability can pose a bottleneck for practical deployment.

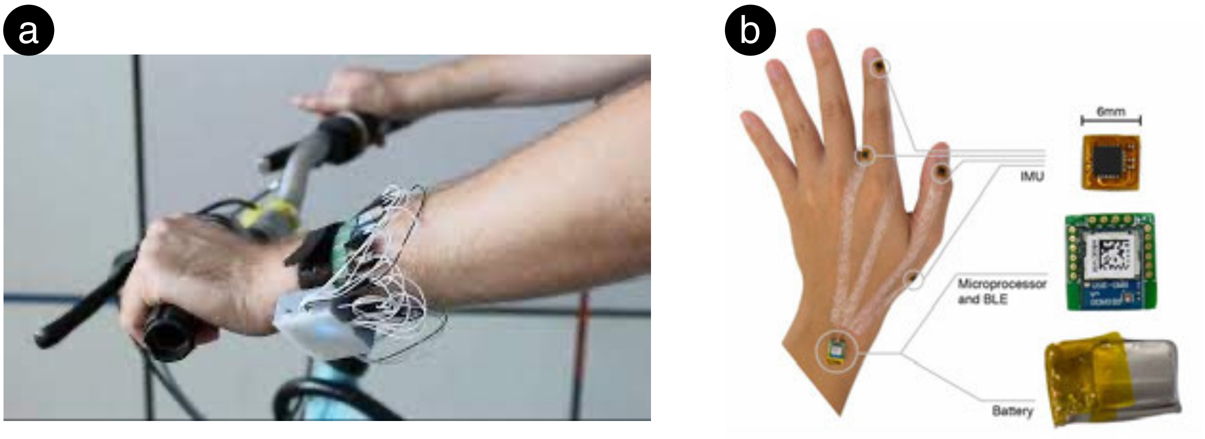


Figure 2.4: (a) WristFlex [74] enabled gestures while holding a bike handle using an array of FSR sensors worn on the wrist. (b) SkinWire [79] proposed a fabrication process of an on-skin hand gestural interface, including a microprocessor, battery, and wireless communication with IMU sensing. (Image courtesy: (a) Artem Dementyev, (b) Cindy Hsin-Liu Kao).

Another approach is based on data gloves that are instrumented with sensors [14, 80, 81]. Despite being able to capture high-fidelity information (as shown in Chapter 5), they are often bulky and hence impede the dexterity of fingers. For a more detailed overview of the different vision-based and glove-based approaches, we refer to [82].

The most closely related approach to our goal of supporting gesture detection in both conditions, freehand and while grasping an object, is proposed using an electromyography band by Saponas et al. [12] and a Force Sensitive Resistor (FSR) band by Dementyev et al. [74] (see Figure 2.4-a). However, the selected grasp variations and the number of gestures are limited due to the lower resolution of the technique. Laput et al. used a smartwatch accelerometer to detect coarse freehand gestures and also demonstrated activity detection [24, 57]. Furthermore, placing an IMU on finger segments has been shown to be effective in capturing subtle finger movements and does not get affected if there is an object in hand [83–85]. Kao et al. [79] used 4 IMUs for tracking index-finger and thumb movements in a fully self-contained on-skin form factor (as shown in Figure 2.4-b). Recently, DualRing [86] presented the usage of two IMUs placed on the thumb and index finger's proximal segment to detect four grippings postures but did not consider any gestures while holding objects. Bardot et al. [87] suggested the usefulness of a smart-ring (embedded with an IMU and a touchpad) for gestures in hands-busy situations. Thus, in order to simultaneously support gestures with freehand and while holding object conditions, we selected IMUs as our sensing technique and present a working system in Chapter 5.

2.3.2 Optimal sensor placement

While the aforementioned works presented a viable technological solution to capturing finger information while holding objects, these do not investigate the optimal sensor placement to fully harness the capability of IMU sensing. Yet, the placement of sensors is as crucial for gesture detection as selecting the appropriate sensing type. This is prominently shown by the findings from Gu et al. [88] and Shi et al. [89] who used a single IMU and determined that touch-contact recognition performance can be strongly increased by investigating the optimal position on different finger segments. Lin et al. [90] used an array of strain gauge sensors to detect finger gestures based on American Sign Language and reported the minimum accuracy of 70.8% can be increased to 95.8% for an identified optimal location. Kubo et al. [91] applied piezo-electric elements to detect thumb and thumb-to-finger gestures, and palm touches and reported the change in accuracy from 90.6 to 96.6% for an optimal location. All these works employed a trial-and-error approach of moving the sensor at different locations, requiring considerable time and effort. We leverage a large dimensional dataset captured using a dense setup of 17 IMUs to avoid the process of repeating manual trials involving the movement of a single sensor at different locations.

Using the principle of compressed (or sparse) sensing, a large body of work has demonstrated that a significantly reduced number of sensors can reconstruct real-time human body pose. Schroder et al. [92] used subspace-constrained inverse kinematics (IK) to demonstrate only one marker per degree of freedom is sufficient to capture the articulations of hand, as shown in Figure 2.5. Andrews et al. [93] fused the data from sparse set of six IMUs and five optical markers to reconstruct the human-body motion using a physics-based framework and an inverse dynamics solver. Huang et al. [94] reconstructed human pose using 6 IMUs by synthesizing IMU data from MoCap datasets and modeling temporal information using a bi-directional recurrent neural networks architecture. More recently, Eckhoff et al. [95] proposed a sparse and magnetometer-free IMU tracking of double-hinge joint systems with non-parallel joint axes using kinematic constraints. However, as mentioned by Brunton et al. [96], reconstruction and classification are two different

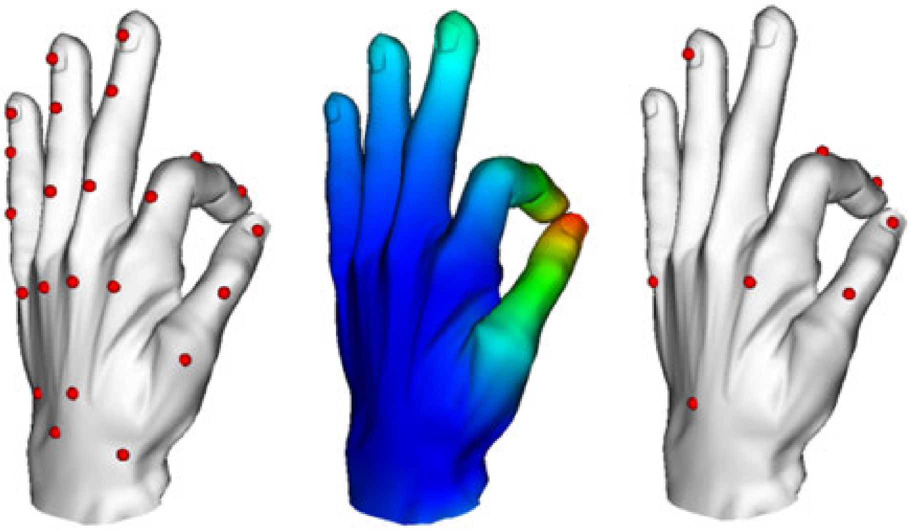


Figure 2.5: Schröder et al. [92] proposed an approach using subspace-constrained IK to reconstruct hand articulations with sparse optical markers. (Image courtesy: Mario Botsch).

problems.

While some work exists that uses sparse representation for gesture classification, they mainly rely on visual data and in free hand scenarios. Poularakis et al. [97] used the video sequences from the 10 Palm Graffiti Digits dataset and proposed a sparse representation-based classifier approach. Mantecón et al. [98] used the depth imagery data from Kinect 2 sensor and introduced a dimensionality reduction technique on the large depth-based feature descriptor. To the best of our knowledge, our presented method in Chapter 5 is the first that presents a computational method for identifying a sparse layout for gesture classification using IMUs. Our method is not only capable of considering gestures performed in freehand and grasping conditions but also rapid enough to select customized layouts supporting user-defined requirements of gestures, objects, and location constraints for sensor placement.

2.4 Computational design approaches

The idea of reducing the effort involved in implementing gesture recognition systems has received considerable attention in HCI. Wobbrock et al. [99] proposed the $1 recognizer for rapid prototyping of gesture-based interfaces. Long's Quill [100], a pen gesture system, enables users to create pen gestures by example. EventHurdle [101], M.Gesture [102] and Mogeste [103] enable users to compose custom gestures on mobile devices. Gesture Coder [104] helps developers incorporate multi-touch gestures into their applications using the developer's demonstration on the tablet's touchscreen. Note that gestures on mobile devices have received particular attention in this trend. Our work goes beyond the restriction of holding smartphones or tablets and focuses on enabling gesture detection when the users' hands are busy during everyday activities, like cooking in the kitchen or carrying items on the street.

While there are existing machine learning-based frameworks and platforms for quickly prototyping and debugging various classifiers and implementing custom machine learning pipelines [105–107], they are targeted for programmers and do not consider aspects of interaction design. On the other hand, recent advances in technology have enabled novice users to train and classify custom ML models without the need for programming expertise [108]. However, these majorly address image or audio classification problems (see Figure 2.6). Our main goal in this thesis is to use machine learning as a design material [109] and enable designers to rapidly create efficient gesture recognition without needing expertise in ML and programming. Motivated by the challenges of designing a sparse sensor layout, we strive to provide designers with a computational tool that abstracts from the complexity of multiple factors (choice of gesture, object, and location constraint), which are conventionally tuned by manual efforts and require technical skills. In Chapter 5, we present a GUI-based tool to help designers and engineers select the optimal sparse sensor layout to facilitate always-available input in hands-free and busy scenarios.

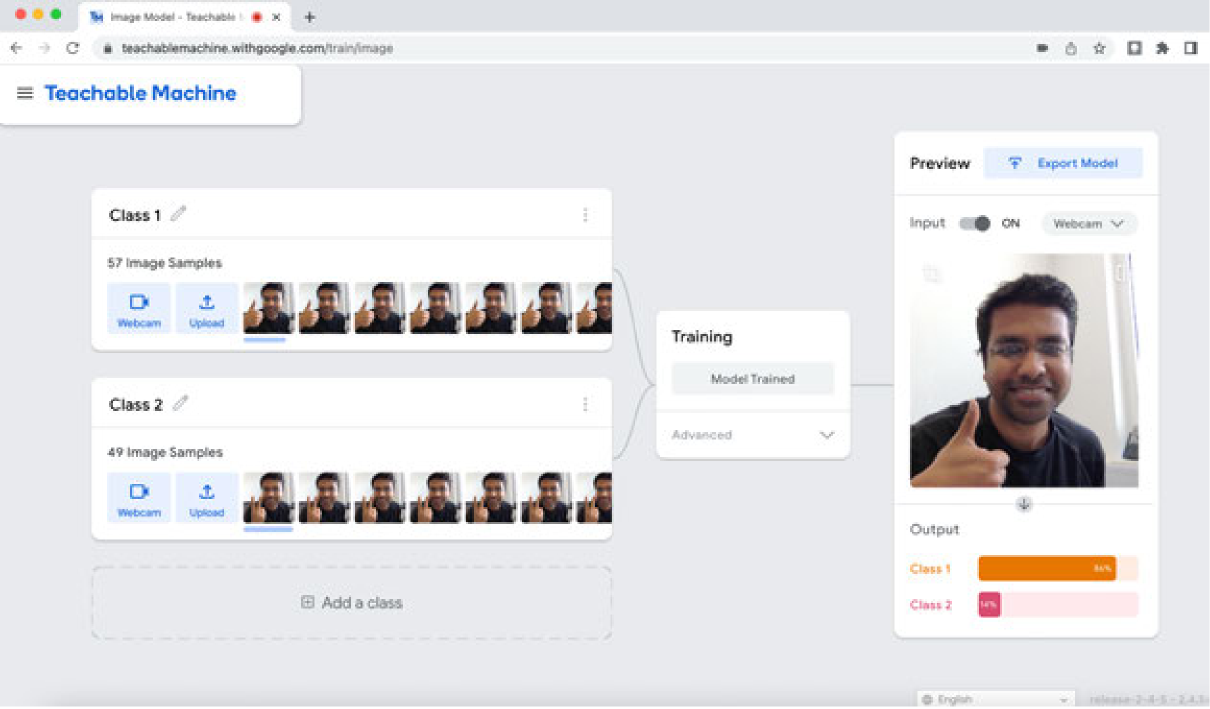


Figure 2.6: Google's Teachable Machine [108] provides machine learning classification using images from a webcam (screenshot attached with permission from Michelle Carney).

CHAPTER 3

User-defined microgestures while grasping everyday objects

Gestural user interfaces typically require users to have at least one hand free, e.g., moving a mouse, touching a screen, or performing mid-air gestures. Consequently, it is difficult for users to operate computing devices whilst holding or manipulating everyday objects, a challenge that needs to be solved for realizing "always-available input". Most closely related, Wolf et al. [5] presented pioneering first work with an intuitive set of microgestures: subtle and rapid finger movements compatible with holding objects designed by investigating three objects. However, we still lack a systematic understanding of how microgestures relate to the different grasps and objects we encounter in our day-to-day lives.

This thesis addresses the above challenge (**RQ I:** How does the multitude of grasp types and object geometries affect users' choice of microgestures?) by introducing *Grasping Microgestures*[1]. We conducted the first elicitation study of microgestures with handheld objects to systematically investigate the effects of grasp and object sizes on the gestures conceived by end-users. We took a holistic approach and probed gestures on objects encompassing all six grasp types from a well-established grasp taxonomy [15]. Additionally, we used two size variations for each grasp type for a more detailed analysis. Besides analyzing their choice of gestures, we also examined information related to the constraints imposed by holding objects, such as the finger used and the location of performing the gestures. Overall, we performed an empirical analysis of microgestures performed while the user held an object in hand and analyzed over 2,400 user-generated microgestures.

Our findings characterize the users' preferred gesture type when hands are busy holding an object. We found that these gestures are largely determined by the referent (command invoked by the gesture), rather than the grasp or object. Interestingly, the choice of

[1] This chapter's contents are based on a publication at CHI '19, which I led as the first author [20]. I created the study design, conducted experiments, and analyzed data (including the idea of using clustering).

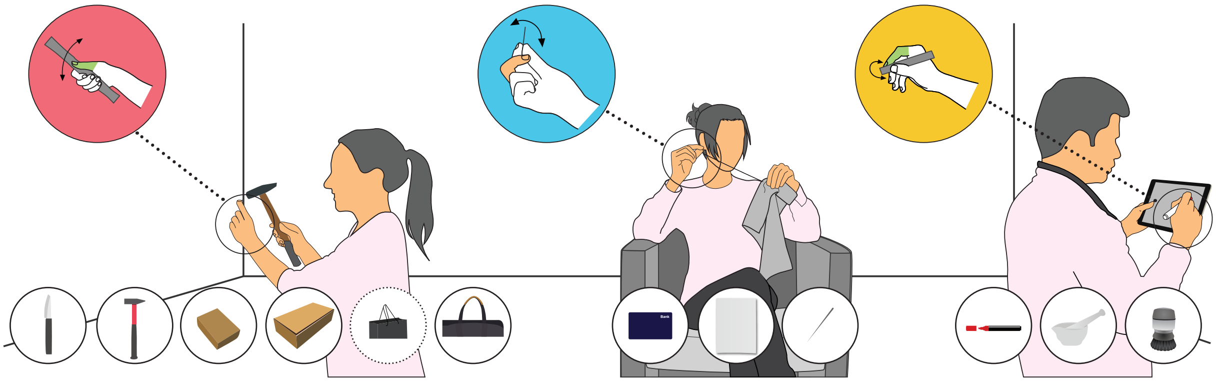


Figure 3.1: Grasping Microgestures enable direct and subtle interactions with computer systems while holding an everyday object. This chapter presents empirical results from an elicitation study with varied objects, investigating the effect of grasp and object size on user's choice of microgestures, preferred locations, and fingers used.

fingers and gesture location is strongly influenced by the handheld object's grasp and size. We extend the original elicitation method by proposing statistical clustering of users' elicited gestures. This approach facilitates finding previously undiscovered patterns through a data-driven interpretation. Furthermore, it identified similarities among different geometries and ultimately allowed us to present three main cluster sets of gestures that cover interactions for all 12 varied objects. We ultimately derive design implications that guide designers of microinteractions in choosing microgestures compatible for use with handheld objects.

This chapter first describes the method of our study design in Section 3.1. Section 3.2 presents the results, including agreement score, distribution of fingers' usage, qualitative analysis, and the clustering approach. Section 3.3 presents a consensus gesture set, followed by the implications for design in Section 3.4. Finally, in Section 3.5 and 3.6, we provide limitations and conclusions based on this chapter, respectively.

3.1 Method

To investigate how users perform microgestures while they are holding objects using various grasps, we conducted an elicitation study.

3.1.1 Participants

20 healthy participants (10m, 10f, mean 26.2y; median 25y; 2 left handed) were recruited from different professional backgrounds (arts, engineering, law, psychology) and various cultural backgrounds (Western Europe, Middle East, India, China, USA). Participant compensation was 20 euros.

3.1.2 Apparatus

Following the method proposed by Wobbrock et al. [40], we intentionally refrained from using any sensing technology so as not to bias the user's response by restrictions imposed by equipping everyday objects with sensors. Participants used passive handheld objects. No additional feedback was provided. The entire session was video recorded.

3.1.3 Referents

Our list of referents is informed by [4, 40]. In total, we selected 10 referents that comprise discrete (*select*, *delete*), binary (*accept/reject*, *next/previous*) and continuous (*increase*, *decrease*, *move*, *rotate*) commands. We kept the set of referents compact, first because microinteractions are commonly used for a small set of simple and quick commands that do not disrupt a primary activity, and second to keep the study feasible despite the number of conditions, which was considerably larger than in typical elicitation studies from prior work.

3.1.4 Grasps

We based our grasp conditions on Schlesinger's seminal classification of six prehensile postures that account for variations in object shape, hand surfaces and hand shape [15, 29]. This classic taxonomy is frequently used in prior work [12, 16–19, 28]. The grasp conditions are:

- *Cylindrical*: for holding cylindrical objects, such as a coffee mug.
- *Palmar*: for grasping with palm facing the object, such as grasping a book.
- *Hook*: for grasping a heavy load such as a bag.
- *Lateral*: for grasping flat objects such as paper.
- *Tip*: for grasping small objects such as a pen.
- *Spherical*: for holding spherical objects such as a ball.

3.1.5 Object Size

We hypothesized that within each grasp type, the size and weight of the object would affect the grasp and hence the set of microgestures that can be performed. We performed a pilot study with two interaction designers who were asked to perform any microgestures they could think of on objects of largely differing weight (ranging from a feather-weight styrofoam ball to a 10 kg dumbbell) and largely differing size (ranging from tiny needle to a 75 cm yoga ball). The results of this pilot study indicated that size has a strong effect on microgestures. To give only one example, while holding a cylindrical object of small diameter, the user can perform actions such as snapping around the object or touching his fingertips. These are not possible with larger diameters. We found that weight has a much less strong influence on the microgestures that can be performed, as long as the weight allows a user to comfortably hold the object using a single hand. For example, one can tap the same way on a very heavy ball and on a lighter ball.

We therefore decided to investigate variations of object size only and selected a small and a large object for each grasp.

3.1.6 Representative Handheld Objects

We chose a total of 12 handheld objects that represented our 6 grasp conditions as well as a significant variation in size within each grasp. The set of objects is shown in Fig. 3.2. To identify representative objects that cover varied environments, two interaction designers have iteratively compiled a list of objects, selecting objects from the literature [5, 110] and adding further ones from everyday usage. We opted for real-world objects instead of abstract geometrical props to make it easier for participants to conceive gestures they would make in a realistic setting. Our final set of objects contains: *knife* and *hammer* for cylindrical graps; *small cardboard box* and *large cardboard box* for palmar grasp; *bags with small and large handle* for hook grasp; *credit card* and *A4-size paper sheet* for lateral grasp; *sewing needle* and *marker* for tip grasp; *pestle* and *scrubber* for spherical grasp.

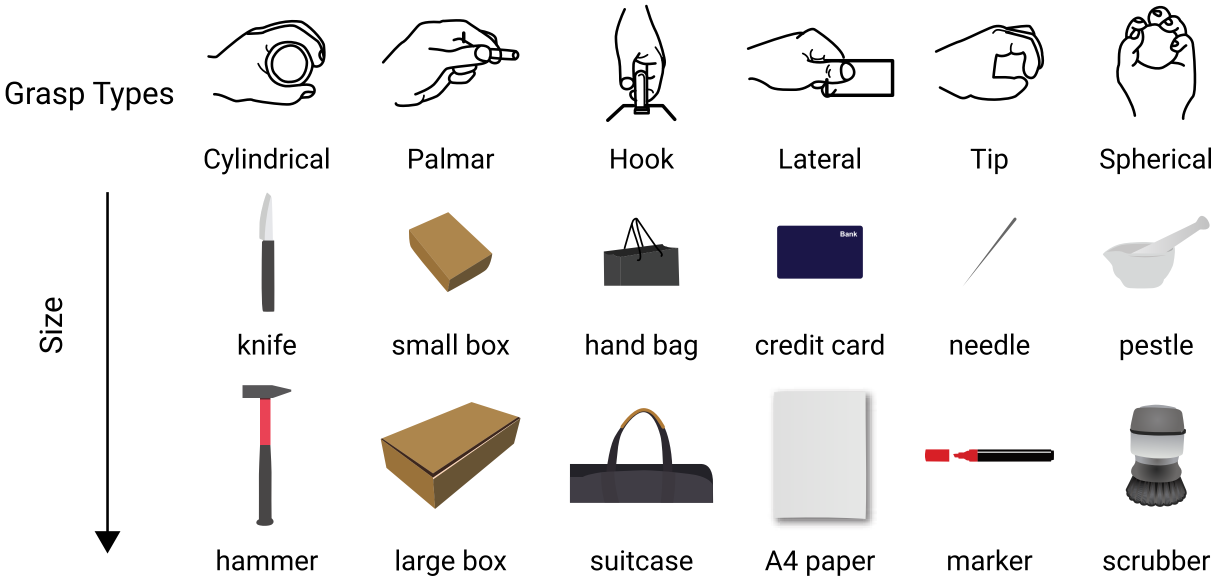


Figure 3.2: Selected grasps and corresponding objects for small and large object sizes.

3.1.7 Task and Procedure

We used a within-subject design. The order of referents, grasps and object sizes was randomized. Participants elicited gestures while standing. First, we chose one of the 12 objects (in random order). The participant was given the object that represents the grasp and object size condition of this trial and was asked to naturally hold it steadily in the dominant hand. For each object, we then presented all 10 referents one after another, in a random order. For each referent, the participant had to make a microgesture using the same hand that was holding the object. To reduce legacy bias, we applied *priming* [111] by informing participants about the potential of such 'Grasping Microgestures'. In addition, we ensured today's computing technology was neither used nor visible during the study: names of objects and referents were presented on paper slips, and we asked participants to place their personal devices out of sight.

In a few cases, the participants chose a different grasp than the one to be tested in the trial. Then the experimenter asked the participant to present a second gesture using the correct grasp. We also asked for a second alternative if the proposed gesture involved rotation or movement of the object. This was taking into account that in some real-world environments it would not be possible to move or rotate the object (e.g., a glass full of water or a power tool).

For each participant, the experiment took approx. 3 hours and was conducted in two sessions of 1.5 hours each.

3.1.8 Analysis

Overall, we elicited 10 (referents) x 6 (grasps) x 2 (object sizes) x 20 (participants) = 2,400 microgestures. An additional 131 microgestures were performed in case of change in grasps, object movement or rotation, as described above. This gave a total of 2,531 gestures. We used descriptive labeling, chunking, and phrasing [112] for our data analysis. We analysed more than 50 hours of video recording and manually annotated each proposed gesture with its properties: which type of action was performed (e.g., tapping, sliding, pressing), direction (if applicable), count (e.g., 2 for double-tap), finger(s) used (including phalanges of the fingers and the thenar and hypothenar eminences), location type (on object, on body, or in air), location on object faces (similar to [6]). The labels for action type and location type were iteratively refined using an open coding approach.

3.2 Results

In this section, we present results of the elicitation study. We analyze agreement between participants and analyze the properties of the proposed microgestures, including action types, location of interaction and finger usage. The results show that microgestures strongly depend on the type of grasp and the size of the handheld object, as these offer different affordances and constraints. We are able to show that all 12 object conditions can be clustered into four types, for each of which we present a consolidated consensus gesture set.

3.2.1 Agreement Rate

To identify the level of consensus between participants' proposals, we calculated agreement rate between participants using the AGreement Analysis Toolkit (AGATe) and the modified agreement rate introduced by Vatavu *et al.* [113]:

$$AR(r) = \frac{|P|}{|P|-1} \sum_{P_i \subseteq P} \left(\frac{|P_i|}{|P|} \right)^2 - \frac{1}{|P|-1} \tag{3.1}$$

We considered participants to be in agreement if they proposed a gesture of the same action type and the same properties, for instance same direction of swiping or same number of taps. This resulted in 18 unique gestures. Agreement rates were calculated individually for each grasp and object size.

The results are shown in Figure 3.3. Agreement rates ranged from 0.049 (low agreement, AR $\leq$ 0.1) to 1.000 (very high agreement, AR $>$ 0.5). The mean AR across all objects and referents was 0.281 (SD = 0.19), which can be qualified as medium agreement (0.1 $<$ AR $<$ 0.3). This range of agreement is comparable with those reported in prior work involving hands as a primary input [4, 45, 49].

3.2.1.1 Agreement rate among different referents

We observed considerable variation in agreement rates for different referents, as commonly reported in prior work. Participants appeared to agree more for commonly used operations like Select. This can be explained not only by a stronger legacy bias, but also by the relative ease of finding a simple mapping for referents such as tapping for select. We also observed higher consensus for commands related to physical actions (Move, Rotate), for which most participants proposed gestures that involve directional movement. In contrast, we observed lower agreement rates for critical commands such as Delete and Reject. Many participants intended to avoid false activation of such critical operations and hence tried to make unique suggestions.

3.2.1.2 Agreement rate among different grasp types

Our results reveal that agreement rates vary among different grasp types. Palmar and Cylindrical grasps show higher agreement than the remaining grasps. This finding might

REFERENT	OBJECTS													
	cylindrical		palmar		hook		lateral		tip		spherical			
	small	large	small	large	small	large	small	large	small	large	small	large	MEAN	STDEV
select	0.300	0.321	0.300	0.342	0.276	0.242	0.314	0.321	0.321	0.405	0.219	0.347	0.31	0.05
accept	0.195	0.143	0.224	0.142	0.174	0.205	0.152	0.147	0.174	0.113	0.105	0.194	0.16	0.04
reject	0.137	0.138	0.065	0.065	0.147	0.148	0.147	0.258	0.347	0.119	0.087	0.152	0.15	0.08
delete	0.072	0.110	0.090	0.065	0.071	0.049	0.087	0.083	0.073	0.083	0.082	0.057	0.08	0.02
next	0.179	0.105	0.168	0.132	0.084	0.189	0.147	0.086	0.142	0.105	0.110	0.081	0.13	0.04
previous	0.174	0.137	0.158	0.174	0.100	0.200	0.116	0.069	0.162	0.074	0.132	0.071	0.13	0.04
increase	0.586	0.637	0.432	0.568	0.186	0.290	0.257	0.242	0.248	0.437	0.479	0.374	0.39	0.15
decrease	0.732	0.563	0.390	0.584	0.179	0.333	0.363	0.247	0.300	0.426	0.437	0.374	0.41	0.15
move	0.323	0.602	0.589	0.814	0.652	0.478	0.320	0.648	0.317	0.524	0.468	0.344	0.51	0.16
rotate	0.514	0.652	0.814	1.000	0.510	0.729	0.484	0.447	0.308	0.431	0.241	0.308	0.54	0.23
MEAN	0.32	0.34	0.32	0.39	0.24	0.29	0.24	0.25	0.24	0.27	0.24	0.23		
STDEV	0.22	0.24	0.24	0.33	0.19	0.19	0.13	0.18	0.09	0.19	0.16	0.13		

Figure 3.3: Agreement rates for all referents, shown individually for grasps and object sizes.

be related to the constraints imposed by these grasps, which restricted finger movement more considerably than in other grasps. Object size had a less considerable influence on agreement rates.

3.2.2 Action Types

To understand what actions the proposed microgestures contain and how the choice of action depends on the referent and on the handheld object, we identified action types and their distribution for referents and objects.

The results are depicted in Figure 3.4(top). They show that the type of action chosen strongly depends on the referent. We identified the following action types:

1. *Tap* (26.1% of all proposed gestures): Participants chose tapping actions most frequently for 3 of the 10 referents (Select, Accept, Delete). For Select, 79.3% of all proposals involved tapping. During the think-aloud session, users mentioned its ease and resemblance to input on touch devices. Participants also leveraged the spatial precision of choosing one specific location of tapping in a some proposals for Accept, Reject, and Delete, as well as for Next and Previous.

2. *Press* (8.2%): Press was among the least performed actions. Some participants intentionally used pressing, as opposed to tapping, as a means to confirm for Select, Accept, and Delete.

3. *Stretch* (9.2%): Some proposals included in-air finger movement, such as pointing with a finger, or stretching out one or multiple fingers. For Reject and for Delete 16 participants proposed stretching out two or three fingers (middle, ring and pinky), as if to flick something away.

4. *Swipe* (37.7%): Continuous actions such as Increase-Decrease and Next-Previous leveraged the fluid, directional as well as continuous nature of swipes. Although all referents were shown in random order, and hence dichotomous pairs of referents were not necessarily presented one after another, participants intentionally made use of opposite direction movements for such dichotomous pairs *("outward as increase, towards myself is decrease" [P10])*. Participants also acknowledged that object geometry plays an important role in helping map directions.

5. *Draw* (16.4%): We classified all non-linear swipes as Draw. Participants used this action in more than 80% of the proposed gestures for Move and Rotate, leveraging intuitive spatial mappings. For instance, a circular sliding motion was used for rotate, while directional movements similar to input on a trackpad were used for Move. 6% of the proposed gestures for Delete were a 'cross' symbol.

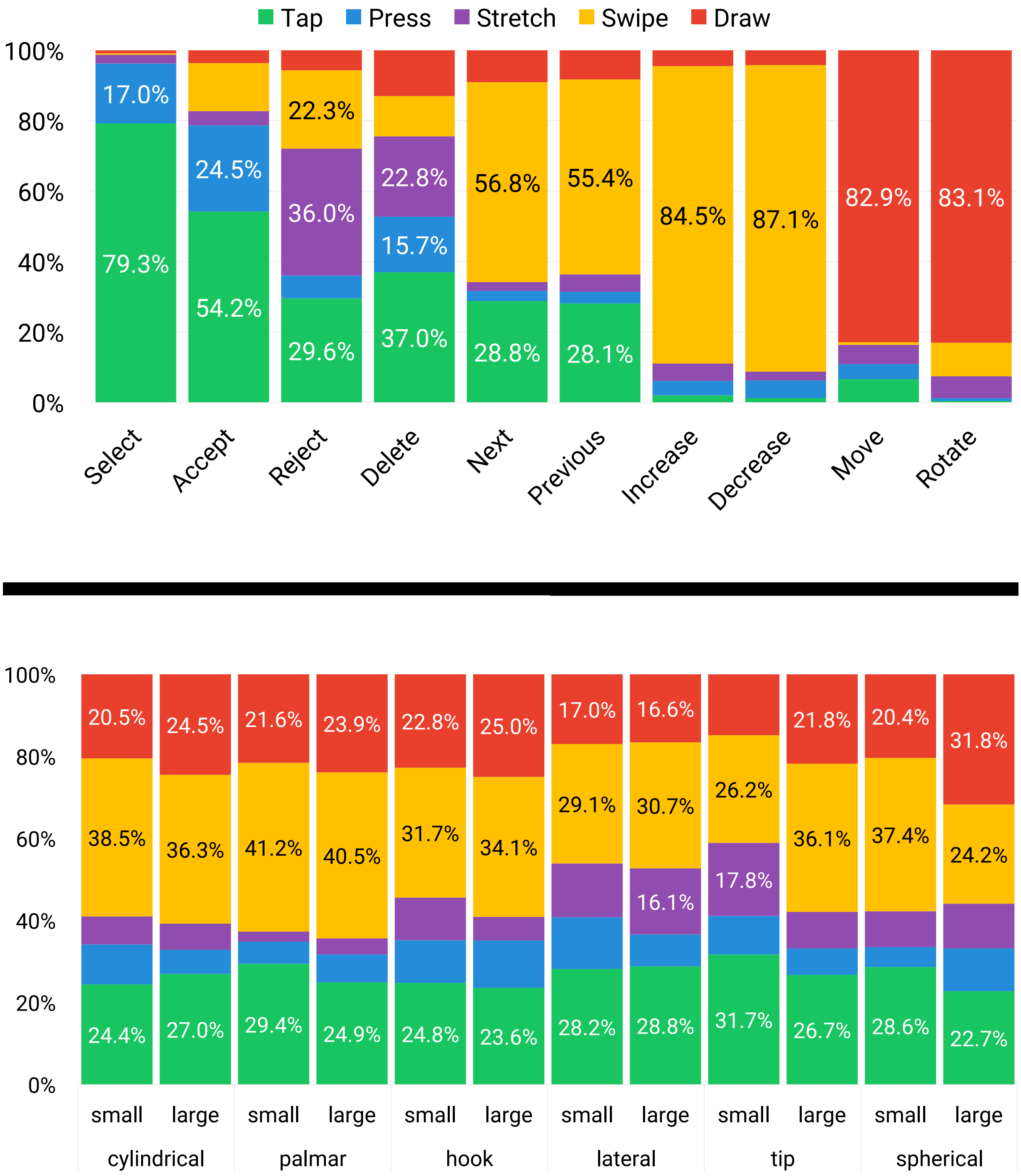


Figure 3.4: Action distribution for Referents (top) and Objects (bottom).

In addition to these types of action, a very small number of proposals involved changing the grasp (0.9%), moving the object (0.5%), or rotating the object (0.8%). As these were very rarely proposed and would not be compatible with all objects, we do not recommend using those.

Contrary to distribution across referents, grasp and object size did not considerably affect the choice of action type, as shown in Figure 3.4 (bottom). A few minor exceptions are notable, however. Spherical grasp with large object (Scrubber) showed the highest percentage of draw actions, which represented almost one in three proposed gestures of this condition. We observed that the thumb movement is restricted, however, the index finger can move easily over the large surface and draw gestures, similar to the posture of holding a computer mouse. In-air gestures were performed mainly with grasps on objects where some of the users' fingers were not involved in holding the object and hence free to be moved in mid-air. This is visible in case of lateral grasp (13.1% and 16.1%) and tip grasp with the small sewing needle (17.8%).

These findings confirm empirical findings of prior work that investigated designers' rather than users' mental model. They also extend to a larger set of grasps and objects and for the first time quantify distributions.

Overall, we can conclude that the choice of action type is mainly guided by the referent, rather than the grasp or object.

A second central implication of our findings is that the vast majority of proposed gestures uses tapping, swiping or drawing, which are all established multi-touch interactions common on handheld devices. Taken together, these findings suggest there might be a possibility of defining consistent microgestures for handheld objects that use similar actions for all objects while being compatible with a user's established mental model of multi-touch interaction.

3.2.3 Action Location: On-Object, On-Body, In-Air

While action type appears mostly unaffected by grasps and objects, our results show that the handheld object strongly influences the location *where* this action is performed. As we did not put any restrictions on where participants did microgestures with their dominant hand, participants were free to perform those not only on the handheld object itself, but also on their own hand or fingers, or in mid-air.

As shown in Figure 3.5, the location used for interaction depends on the size of the object. This plot arranges objects based on the proportion of microgestures that were done *on* the object, rather than on the body or in air. (Note that in all these cases the participant kept holding the object.) It becomes apparent that with increasing size of the object, a higher proportion of gestures are on the object. On the box objects, around 90% of all gestures have been performed on the object itself. Some users commented that

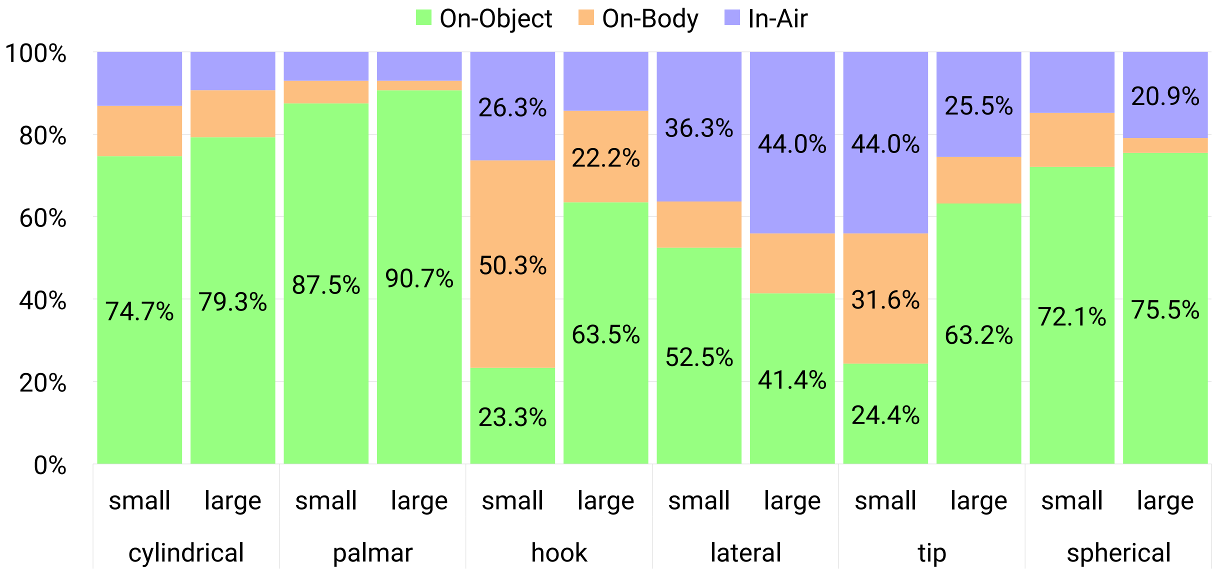


Figure 3.5: Action location for each object.

making gestures on the large box is similar to using a touchpad. On the contrary, as small objects do not offer large surface real-estate for performing gestures, a considerably higher fraction of gestures was performed on the user's own hand or fingers, or in mid-air in case of small objects. The most extreme case, the needle, offers virtually no space for gestures, hence almost 90% of all gestures have been made on the body or in mid-air.

44.0% of the gestures performed using the needle were mid-air gestures due to the lack of surface area on the object. Very few people used their index finger as a pointer. In case of the needle, and also for paper, a common strategy consisted of making touch gestures with a finger on the palm of the *same* hand, like on a touchpad. Some participants were amused once they found out that they can actually touch their palm with the same hand's finger. Once they discovered this affordance, they started leveraging the considerable space for gesturing provided by the thenar region while holding small objects. 12.1% of all proposed gestures for paper and 11.0% for needle used this strategy.

In contrast to self-sustained objects, such as the steering wheel [6], the hand could not be easily moved on our handheld objects to reach distant locations without risking the object falling down. Hence, the majority of microgestures appear close to the position where the object is held. Furthermore, none of our participants used prominent visual landmarks on the object (like a printed logo) for interaction.

3.2.4 Use of Fingers

While prior work on free-hand microgestures has identified frequency rates of finger use [4], we are not aware of any such information reported about microgestures with handheld objects. Taking into account that grasping an object constrains finger movement, it is to be

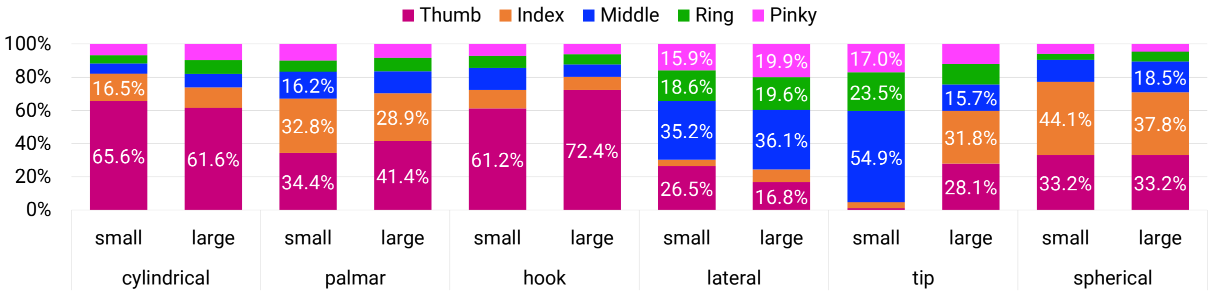


Figure 3.6: Fingers used as an actor for grasping microgestures.

expected that different grasps considerably affect which fingers are used for microgestures. Here, we contribute the first frequency usage for each finger based on different grasps and object size.

The results are depicted in Figure 3.6. They empirically confirm that the grasp considerably affects the choice of fingers for microgestures. We identified two main clusters, based on grasps: For Hook grasp, Palmar grasp, Cylindrical grasp and Spherical grasp, the vast majority of gestures were performed using the thumb or index finger. These grasp types have in common the use of most or all fingers for holding the object. This allows the user to temporarily move the thumb or index finger, while using the remaining fingers to stabilize the hold.

By contrast, for Lateral grasp and for Tip grasp with a small object (sewing needle), the vast majority of gestures were performed using middle, ring or pinky finger, or a combination of those. These grasps have in common holding the object with both thumb and index finger. As their movement was constrained, participants resorted to using the remaining fingers for microgestures. In those cases, the middle finger was most frequently used. For instance, a user might be comfortable using the middle finger to perform a swipe on a digital pen to increase the stroke width. In contrast, using the thumb would create an imbalance in the grip and might lead to dropping the object. However, there were some instances where participants avoided using the middle finger due to social inappropriateness of gesturing with the middle finger, although the gesture would have been easier to perform than with the ring or pinky finger *("[it is] socially unacceptable if I use the middle finger, which is easy to do" [P4])*.

Tip grasp with a larger object (marker) was situated between both clusters, with a fairly even distribution between thumb/index and middle/ring/pinky.

Analyzing the first group in more detail, our data reveal that microgestures in Hook grasp and Cylindrical grasp most frequently use the thumb and rarely use the index finger. Also, Palmar grasp shows a slight preference for the thumb. We believe this is particularly likely because in such cases the object rests against the other fingers and hence the thumb can be easily released from the object for interaction. In contrast, Spherical grasp and Tip grasp (large object) make more equal use of the thumb or index finger.

Similar to the findings by [6], our data show that the choice of finger used to perform the gesture is almost unrelated to the associated command. Contrary to [4], participants did not complain about not remembering the exact finger. Our assumption is that this is because of the additional constraints present in settings with handheld objects: since the grasp posture restricts the choice of fingers to be moved, it helps users to remember the fingers used for the interaction.

3.2.5 Qualitative Analysis

We used an open-coding approach, with iteratively refined codes, to describe the gestures' properties, such as the type of action, gesture location, and finger details. In addition, we annotated gesture properties with unique observations we made during the study and post-session interviews. Altogether, the analysis revealed several interesting insights about how participants performed gestures. For instance, when performing the gesture on a small object, participants showed a variety of techniques to overcome the limited amount of space on the object. This included slightly adjusting the grasp or retracting fingers that were not involved in a gesture to create additional space on the object for making touch gestures. Participants clarified that techniques like unconsciously bending fingers for creating an interaction surface were inconsequential to the core gesture. Similarly, participants completed linear swipes through diagonal movements when a specific horizontal or vertical movement was not possible on the object geometry (e.g., swiping at the backside of the paper using the middle finger *("I don't have vertical movement...doing it diagonally" [P4])*.

In almost all cases, users performed touch gestures with the center of the fingertip. In some cases, however, like sliding along the pestle, the ulnar (inner) side of the finger was used due to the ease of contact. There were also very few instances where the hand's metacarpus (palm) region was used as an input mainly for Press actions while holding the object. Moreover, participants preferred using the nail instead of fingertips for interactions involving "knocking" on an object. These variations with different finger parts can expand the design space of performing gestures on an object. Several participants commented that they would be willing to repeat the same gesture to allow the system to distinguish the gesture from normal object manipulation and to ensure it is recognized as intended input. While feedback is outside the scope of this study, one participant explicitly stated that he would appreciate getting vibrational feedback as a confirmation the gesture was accepted.

3.2.6 Clustering of Grasps

A major challenge in designing microgestures for use with handheld objects is the large number of grasp types, which is further complicated by additional influencing factors

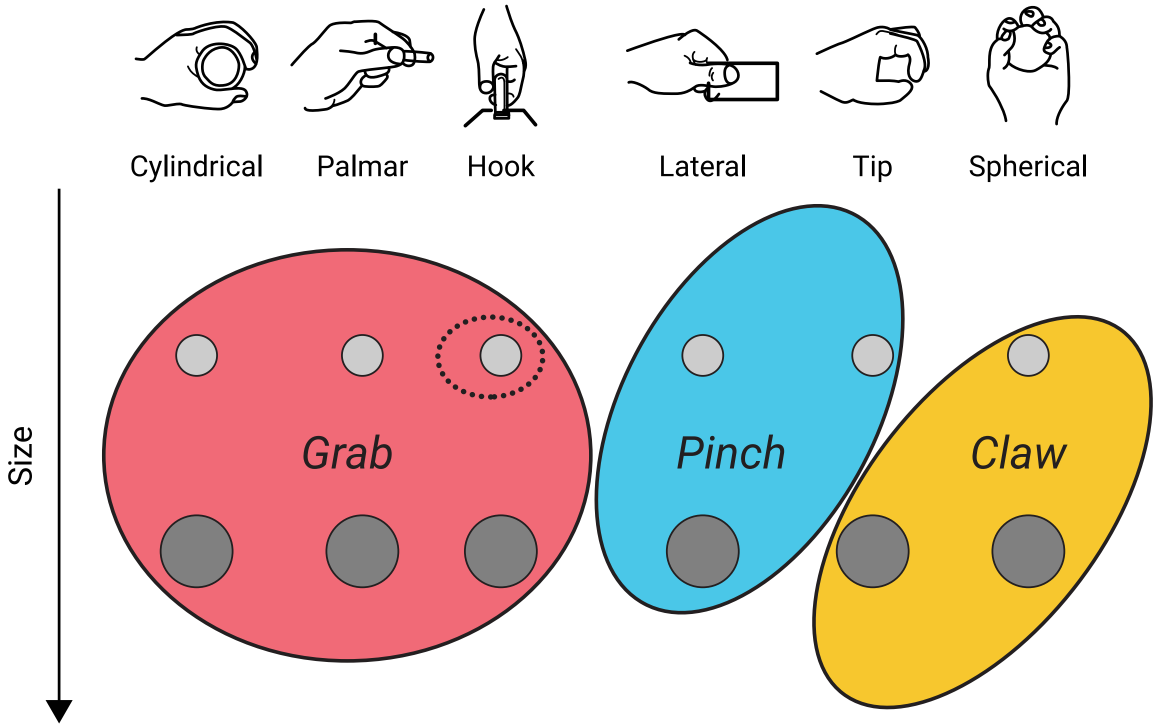


Figure 3.7: 3 Clusters derived from the commonalities of the interaction amongst all 12 representative objects.

such as object size. As it would not be desirable to design individual gesture sets for each condition, we sought to further extend the information provided by the agreement score analysis. In addition to finding consensus gestures for a given referent, we aimed at analytically identifying commonalities among the users' microgesture proposals in different conditions.

While statistical clustering is a commonly used technique in the fields of machine learning and pattern recognition, to the best of our knowledge we are the first to leverage this data-driven approach in an elicitation study to reveal patterns. We used all gesture properties we had annotated in our dataset, including action location, finger use, etc. To avoid a bias, we removed information about the experimental condition (grasp type, object size). We first applied Principal Component Analysis (PCA) for dimensionality reduction of our annotated data. Furthermore, we used the simple yet robust K-nearest neighbor approach for clustering. We employed the elbow method to find the optimal number of clusters (k=5). After analysing the K-nearest output and visualizing the level of separation between clusters, we observed that the frequency distribution of gestures from the same condition across the five clusters showed a significant peak on exactly one cluster for all conditions. Hence, we applied the majority rule to map each condition (grasp x object size) to exactly one cluster. It is worth noting that one of the 5 clusters did not contain any majority vote, and hence became an empty set in our final grouping. The resulting clusters, which we call *Grab-a*, *Grab-b*, *Pinch*, and *Claw* are shown in Figure 3.7.

Cluster Grab-a comprises grasp types where the user's fingers are reaching around the handheld object, allowing thumb or index finger to be moved, while the object is offering considerable surface real estate for interactions. Cluster Grab-b (shown with dotted lines) can be qualitatively explained by the combination of a handle with small diameter and the Hook grasp. This resulted in a unique affordance allowing the thumb to reach around the handle and perform gestures on the user's hand, specifically at the distal phalanx (front) side of the index finger.

Cluster Pinch comprises grasp types that predominantly make use of the thumb and index finger for holding the object. Cluster Claw comprises grasp types that have predominant use of index finger rather than thumb.

Comparing this empirical clustering with the intuitive, conceptual grouping of grasps done by Saponas et al. [12], it is interesting to note that both approaches resulted in three groups of grasps. Our empirical findings confirm the intuitive grouping of Palmar and Cylindrical grasps. Likewise, our findings confirm that Hook grasp forms its own group, however only for small objects. Most important, our findings show that contrary to the grouping proposed earlier, Spherical grasps systematically differ from Palmar and Cylindrical grasps in the use of index finger vs. thumb, and hence should not be grouped together.

3.3 Consensus Gesture Set

We used these three clusters to design consensus gesture sets for microgestures with handheld objects. These are the first end-user driven gesture sets that cover a large range of grasp types and objects.

For each referent, we assigned the most frequently performed gesture, similar to [40]. The gestures for Reject and Delete are grouped together because of a high consensus for this particular action by our participants.

Figure 3.8 shows the final consensus gesture sets. Drawing from the quantitative data and our observations, we suggest that conventional mapping of Tapping and Swiping offers the most convenient mapping for Select, Increase/Decrease, and Next/Previous actions. The press modality has been most frequently proposed for confirmation. Stretching of fingers, used for Reject and Delete commands, require higher user consciousness, reducing the likelihood of any false input. The circular or directional Draw action defined for Move and Rotate provide natural spatial mappings.

Even though we present three consensus gesture sets–one for each main cluster identified in the previous section–it is to be noted that these gesture sets share many features. For each referent, the action type and main action properties are similar in all gesture sets. The first and second gesture sets only differ in use of thumb vs. index finger. The

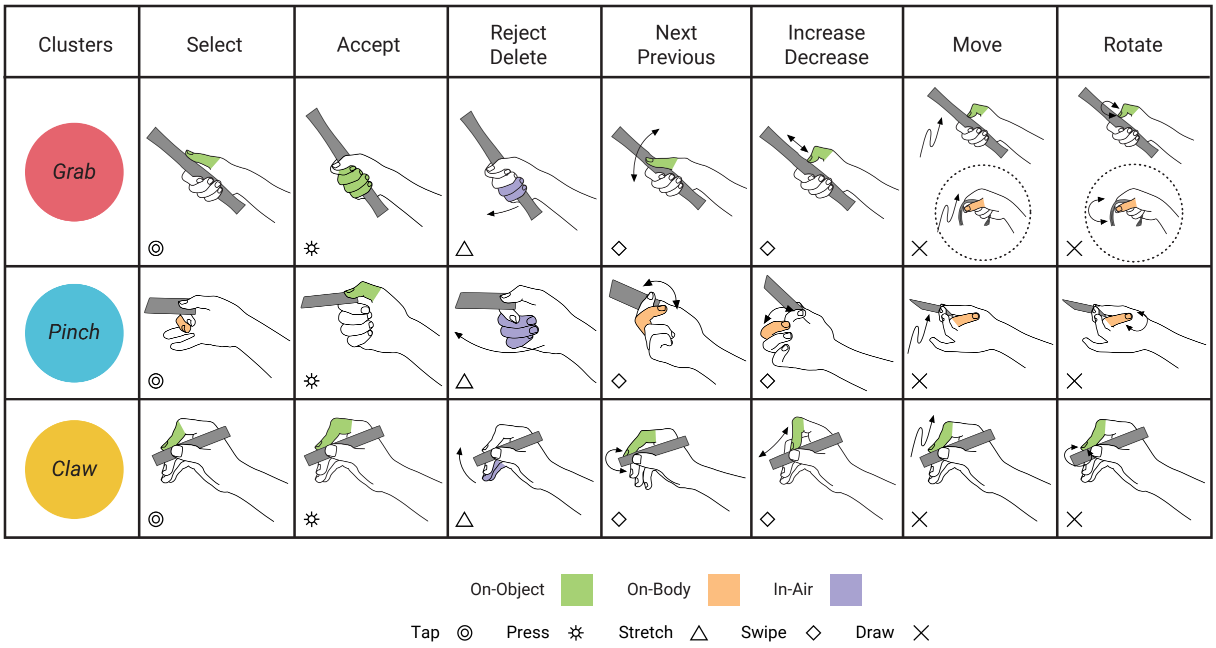


Figure 3.8: Consensus gesture set for all 3 main clusters.

difference between the second and third gesture set is that gestures appear on-body or in-air vs. on-object.

Despite the large variations in grasps and object sizes these user-defined gestures support, we believe these microgestures will be easy to memorize and easy to perform. This is because they build on established mental models of touch interaction, systematically leverage affordances and constraints offered by grasps, and use similar gestures for all grasp types.

3.4 Implications for Design

Based on the results presented in the previous section and qualitative feedback of participants while thinking aloud and during interviews at the end of each session, we derive several implications for design of systems for gestural input.

3.4.1 Microgestures on Everyday Handheld Objects

A central question that motivated our study was to find out how the multitude of grasps and object geometries affect users' choice of microgestures they perform while holding objects. Would designers of applications for mobile computing and the Internet-of-Things have to design a custom set of specific gestures for each type of object? Are there commonalities that allow us to use the same gestures on many objects? The former would be very undesirable from a usability standpoint and would risk frustrating users up to the point of rejecting the new opportunities unleashed by microgestures. The latter would be highly desirable but seemed unrealistic to us before conducting the study.

One of the primary and surprising findings of our study is that three gesture sets are sufficient to cover all 6 main types of grasp and 12 objects in our study. In addition, the gestures are similar for all three sets, as they use the same action types and gesture properties and mainly differ only in what finger is used for making the gesture and whether the gesture is performed on the object, on body or in air. Given these choices are mainly guided by the affordances offered by the object (small or large surface for performing gestures) and constraints of the grasp posture (which fingers can be moved easily while holding the object), we believe users can easily perform the gesture that is compatible with the given object. Ease-of-learning and memorization is further supported by our finding that most gestures build on established touch gestures commonly known from touchscreens. While this might have been strongly influenced by legacy bias, we believe it is a strength of the gesture set, as it is compatible with established mental models of interaction. Our findings further show that miniature objects as small and thin as a needle can be used as an input medium. Participants performed similar microgestures as on other objects, but with more on-body and in-air interaction. Fairly large objects, such as a large cardboard box or a suitcase, can be used for single-handed microgestures, too.

3.4.2 Avoiding False Positives

False positive input is a challenge while interacting with handheld objects, as hand or finger movements that relate to the primary activity might be incorrectly recognized as an input microgesture. While our study design did not focus on this question, our results indicate a number of strategies that participants have used to avoid false positives. Many participants were particularly inventive for gestures that trigger a critical action like Deletion. The most varied actions have been proposed for these referents to ensure they are different from movements that relate to the primary activity *("Normally wouldn't touch down" [P9])*. For instance, one strategy was to intentionally change the grasp while using the marker, and touching the lowest tip part (area with ink). Another strategy was to stretch only the pinky finger while keeping the middle and ring finger in a flexed position. In contrast, during natural movement, stretching the pinky would normally result in at least some stretching of the ring finger as well. Participants went as far as using double or even triple taps, or intentionally touching the sharp area of the knife, to ensure communication of the input gesture is intended. As an alternative to implementing a specific activation gesture or mapping critical functionality to gestures that are hard to perform, we recommend that designers implement an undo functionality that allows the user to undo any previous action that might have been triggered by a false positive. The Reject gesture from our gesture set could be used for this function.

3.4.3 Sensor Placement

The finger and location information provided in this study can be used to inform sensor placement for gesture recognition systems on the handheld object, on the user's hand and fingers, or even both.

Our results show that by only sensing input from the thumb and the input finger, a large majority of all gestures can be sensed for all grasps other than Lateral and Tip with small object. Gestures in Cylindrical and Hook Grasps can even be reasonably captured with the thumb alone. For the Lateral and Tip (small) grasps, sensing input from the middle finger would offer a minimum instrumentation. While previous studies identified the pinky as the least frequently used finger, we uncovered its unexplored dexterity while holding small objects like needle and credit card.

Participants mentioned the thenar region as a large fleshy area of the palm *("Tap with middle finger on the fleshy part of the palm" [P11])*, similar to the 'touchpad' of laptop and used it extensively as an input surface for touch gestures while holding the needle. Designers of sensing systems should consider capturing input on this area. Only 1 out of 20 participants suggested the use of Shear action, and also Press was rarely proposed. We therefore conclude that in most cases it seems sufficient to capture touch contact alone.

Our location information can also be used to avoid false positives on the object, placing the sensor at a location farther from the place where the object is grasped. Being harder to reach, it is less likely the user would interact on it unintentionally.

3.5 Limitations

In our study, we investigated gestures performed within a short pause during the primary activity. As stated by Ashbrook [37], such microinteractions should take less than 4 seconds to initiate and complete to smoothly integrate with the primary activity. For this work, we opted against gestures that would be performed simultaneously, without stopping the primary activity. The effect of many possible physical primary activities are beyond the scope of this study and should be investigated separately in future work.

To help participants invent realistic gestures, we opted for actual objects instead of abstract geometric props. The choice of objects was centered around providing familiarity with the object. Some participants even commented that they have never thought that such objects they commonly use can be used for interaction. Using realistic objects for the respective grasps implied that there is some natural variation in size and weight of objects. We acknowledge this could be a limitation from a formal experiment perspective, yet we believe it is outweighed by the benefits of being able to cover diverse realistic objects in this exploratory study. The effects of object size and object should be investigated in more detail in future work.

For the sake of comparability among objects, we have used rigid objects. Future work should study how affordances of soft materials might change user behavior. For instance, users might perform more squeezing or pressing actions with soft objects.

To create a more relaxed and creative atmosphere, the participants in our study were not blindfolded, which we deemed important for inventive gesture proposals. While most of the proposed gestures can certainly be performed during eyes-free interaction, we clarify that this is not necessarily guaranteed, as participants were able to look at the site of interaction.

3.6 Conclusion

In this chapter, we presented results from the first elicitation study of microgestures with handheld objects to systematically compare the effect of grasps and object sizes on microgestures conceived by end-users. Our findings revealed a strong influence of grasp and object size on usable microgestures. Furthermore, results from data-driven clustering show that the effect of grasp and object size on microgestures can be reduced to only three clusters. Together with the consolidated gesture set, we have presented findings useful for designing gestural input and recognition systems for situations where a user's hands are busy holding an object. These findings are the first step toward unified microgestures that work across all common handheld objects; we hope they will be useful to both designers and engineers of gestural input systems. With finger movements during daily activities or complex hand-object manipulations, they could get mixed with gestures designed by the end-users. Therefore, in the next Chapter 4, we address the critical question of identifying strategies for avoiding false-positive input during everyday activities.

Chapter 4

Robust microgestures for avoiding false activations

The previous chapter identified Grasping Microgestures as a promising means for enabling interaction while holding everyday objects. Such gestures might be used to replace the wake word in voice assistants (e.g., "Alexa" or "Hey Google") [114], but can also offer benefits in a myriad of applications, ranging from controlling mobile devices when on the move and hands are busy, to controlling systems in healthcare contexts [115]. Although performing quick and convenient gestures on handheld objects is desirable, they might conflict with finger movements that might occur when adjusting one's grip or manipulating the object. As a consequence, gesture recognizers may misinterpret natural finger movements as intended input gestures and trigger unintended commands, such as *false activation*. Many recent approaches to reduce false activations involve delimiter gestures and are designed for a specific device [7, 8, 53, 54]. Such approaches tend to disrupt the user's workflow and lack scalability to the multitude of objects in their daily lives.

In this chapter, we present *SoloFinger*[1], a concept to address the problem of false activation when holding or manipulating everyday objects (**RQ 2:** How to avoid false activations in gestural input while handling everyday objects?). It is based on the observation that fingers tend to be static or move simultaneously when holding and manipulating objects. Thus, an *extended* yet comfortable movement of a *single* finger is rare (see Figure 4.1a and 4.1b). This allows the design of simple and robust microgestures that are applicable to diverse grasp types, object geometries, and everyday tasks. To methodologically validate this in the context of hand actions and gesture design, we conducted extensive user studies and performed a series of data-driven analyses. As a

[1] This chapter's contents are based on a publication at CHI '21, which I led as the first author [21]. I conceived the initial concept of SoloFinger and refined it with my co-authors, designed the dataset collection studies, built the OptiTrack setup, and led the data analysis. Additionally, I started and led the collaboration with researchers from the University of Copenhagen.

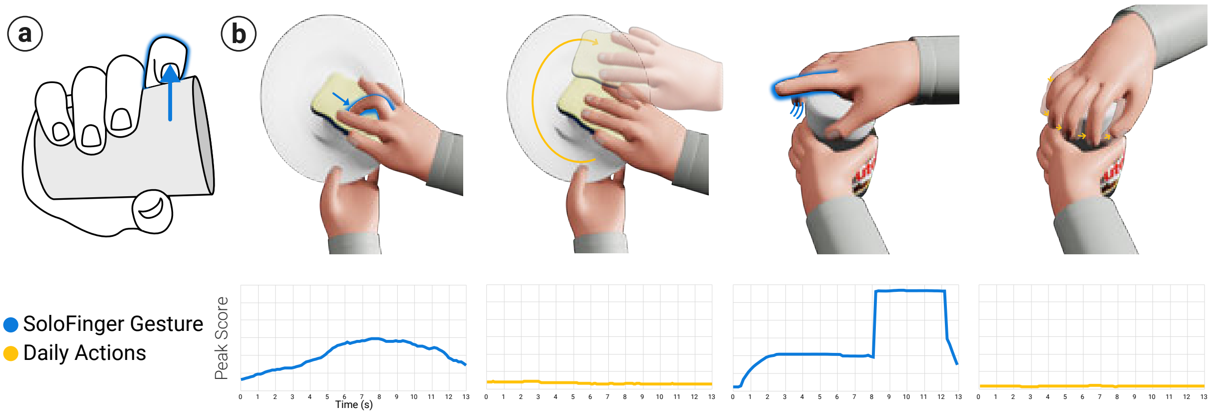


Figure 4.1: (a) The SoloFinger concept: while grasping an object, one can perform a single-finger microgesture while other fingers stay idle. (b) These easy and rapid-to-perform gestures exhibit a distinct movement signature, which is not present during daily actions. This yields a robust gestural input compatible with versatile object geometries and actions.

result, we recommend 7 types of SoloFinger gestures performed with the thumb, index, or middle finger, offering overall 21 interaction options. We collected 7,488 gesture trials. We systematically analyze these gestures as well as a pre-existing dataset comprising 933 trials with daily hand-object actions. To produce findings that generalize beyond a specific classifier implementation and can be interpreted by humans, we opted for a simple white-box classifier, based on thresholding. The results show that SoloFinger microgestures performed on 36 objects can be recognized with an average precision of 100% and recall of 88% (SD = 7) over three primitive gestures. We also show the technique's resilience to false activation on the held-out dataset, which triggers false activation in only 51 out of 933 trials of actions performed with 36 objects. Notably, no false activation was found for 23 actions, while most cases of false activation occurred on extremely deformable or very small objects.

Finally, we demonstrate a proof-of-concept system with a commercially available virtual reality glove and a random forest classifier. This implementation can detect 7 types of SoloFinger microgestures performed with the thumb, index, or middle finger. Classification without knowledge of the held object shows an overall accuracy of 86%, with a very low number of false activations (17 out of 800 trials). When the held object is known, the accuracy further increases to 89%, without any false activation in the collected dataset.

We release two datasets of SoloFinger gestures performed by a total of 21 participants and captured using an OptiTrack optical motion capture system and a virtual reality data glove for 36 and 5 actions, respectively. This fills a gap in the existing literature by providing data about finger gestures while grasping. The datasets are available at: *https://hci.cs.uni-saarland.de/research/solofinger/*

In the rest of the chapter, we first describe the SoloFinger concept. Section 4.2 introduces the two datasets: SoloFinger (captured by us) and Daily Hand-Object Actions

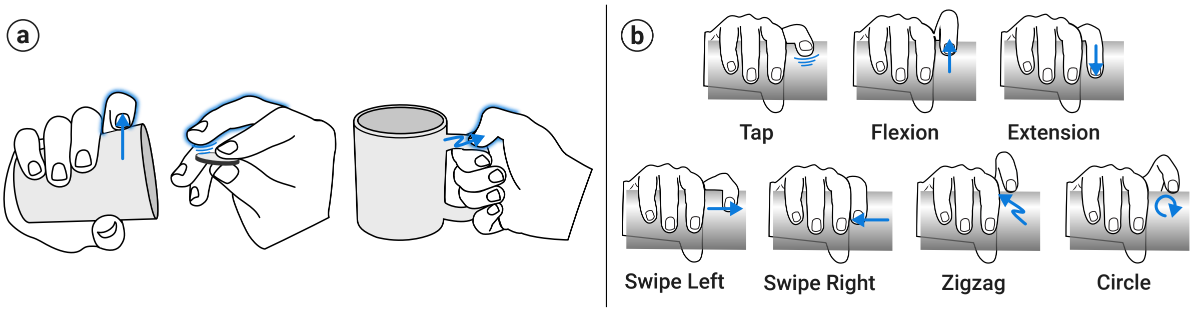


Figure 4.2: (a) SoloFinger microgestures are performed with a single finger on an object, while holding it. (b) Tapping or different directional movements define unique gestures that can be performed either with thumb, index, or middle finger.

(pre-existing [116]) used in our extensive analysis. We provide the validation of our concept in Section 4.3 along with a peak score metric to initially used to compare both datasets, effects of finger individuation, and recommendation on the set of fingers suitable with the concept. Section 4.4 reports on the SoloFinger gesture recognition rate and the evaluation of false activations with both datasets. A proof-of-concept system utilizing commercially available hardware and a multiclass classifier is presented in Section 4.5. Discussion and limitations are described in 4.6. And finally, conclusion for this chapter is outlined in Section 4.7.

4.1 SoloFinger Concept

The sophistication of the human hand allows for dexterous hand-object interactions. Our fingers hold objects using a wide variety of grasps, and while manipulating objects our fingers move in versatile ways and diverse configurations. Due to this impressive richness of movement patterns, it is challenging to define unique gestures applicable across different grasp types and object geometries, yet mutually exclusive of everyday actions. We introduce the concept of SoloFinger microgestures that aim to stand out from everyday hand motions, hence reducing the likelihood of false activations.

SoloFinger microgestures are conceptually based on the observation that during everyday hand-object interaction, multiple fingers tend to move concurrently, whereas it is rare that a single finger moves extensively on the object while all others stay idle. This observation was informed by findings that finger movements tend to be highly correlated during object manipulation [117, 118]. Our work leverages this phenomenon. We ground our findings on objects that do not contain movable elements, such as mechanical buttons or sliders.

A SoloFinger gesture (Figure 4.2b) involves *moving a single finger by a considerable yet comfortable extent*, while *all other fingers remain static*. It is performed *while holding an object, with the same hand*, and *on the object* itself. SoloFinger gestures are not limited to any specific finger. We recommend using the thumb, index, or middle finger, as the ring and pinky fingers were shown to be less robust and also subjectively less preferable.

This generic approach allows for defining diverse microgestures. For instance, these comprise tapping, moving a finger forward, backward or sideways, or moving in advanced patterns, such as drawing a circle. Figure 4.2 (b) depicts the gestures we used in our studies. Performed with either thumb, index, or middle finger, this leads to a total of 21 gesture options we have investigated. However, more SoloFinger gestures can be conceived.

In the following, we will conceptually and practically validate the feasibility of our proposed concept. Our conceptual analysis is based on two datasets that we present in the next section. It validates two key assumptions that underlie SoloFinger gestures: extensive single-finger movements are unlikely to happen during everyday hand-object actions, and SoloFinger gestures are compatible with holding diverse types of objects in diverse grasps. Next, using a simple white-box classifier, we investigate the principled feasibility of gesture classification and show that SoloFinger gestures create little false activation during diverse everyday actions. Finally, we demonstrate the practical feasibility by presenting a proof-of-concept implementation that uses commodity hardware.

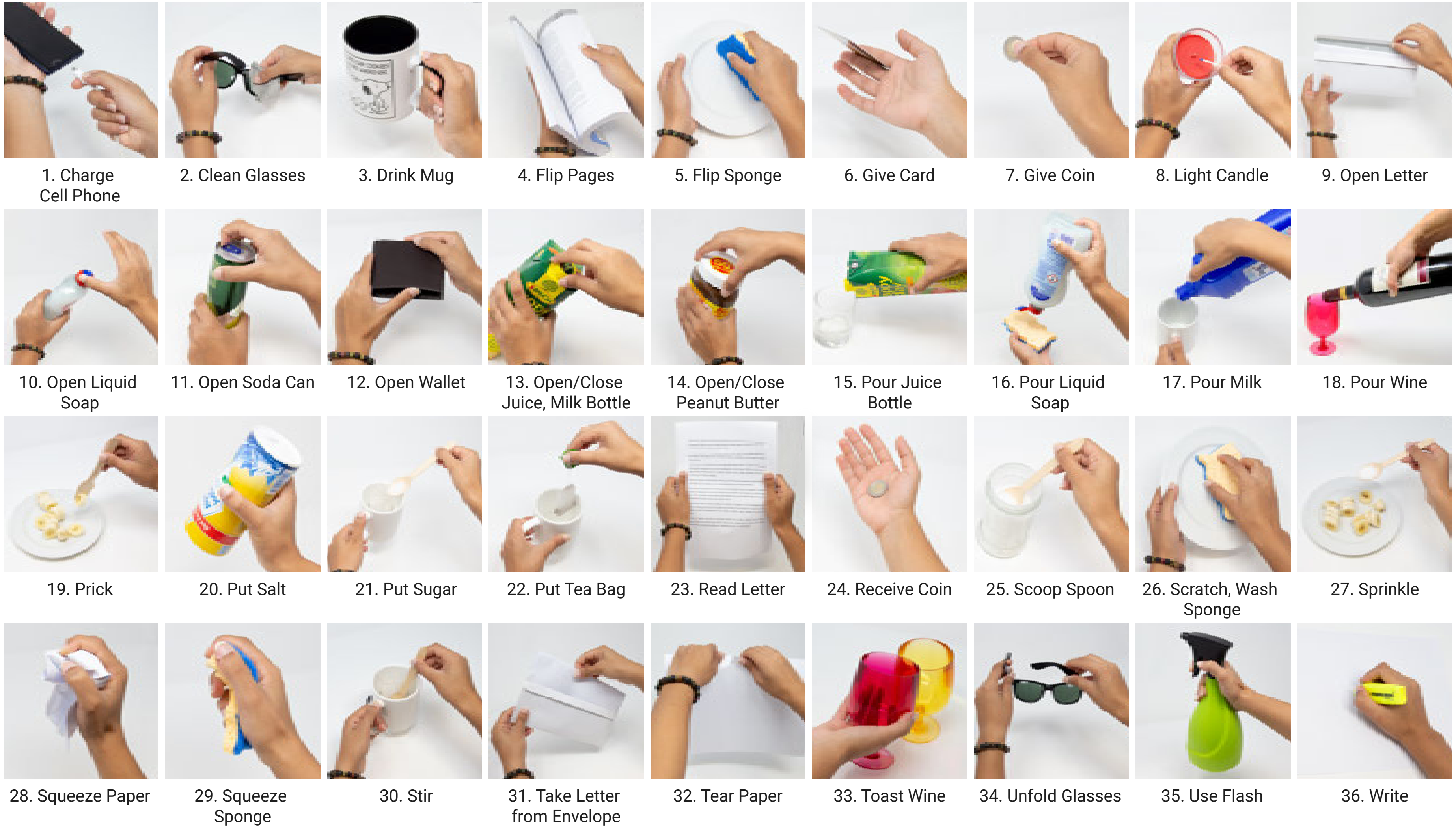
4.2 Datasets for Daily Hand-Object Actions and SoloFinger

We use a data-driven method to systematically validate our concept using two datasets: a baseline dataset offering extensive coverage of everyday hand-object interaction, and a dataset of SoloFinger gestures that end users performed naturally using a wide set of grasps and actions.

4.2.1 Dataset with Daily Hand-Object Actions

We base our analysis of everyday object manipulation on a baseline dataset made available by prior research. Garcia-Hernando et al. [116] created the first benchmark dataset that provides precise information about hand joint positions and angles during an extensive range of hand-object interaction. It comprises data from a diverse set of 45 everyday object manipulation actions, performed with 26 objects. Data were captured using high-frequency magnetic sensors to avoid any obstruction between finger contact and object surface. Information about hand joints and fingertips was then derived using inverse kinematics. The dataset contains 105,459 RGB-D frames with 3D location of each of the 21 joints of a hand model.

We use this dataset to verify our assumption that single-finger movements are rare while grasping an object and for assessing false activations caused by SoloFinger gestures.



1. Charge Cell Phone

2. Clean Glasses

3. Drink Mug

4. Flip Pages

5. Flip Sponge

6. Give Card

7. Give Coin

8. Light Candle

9. Open Letter

10. Open Liquid Soap

11. Open Soda Can

12. Open Wallet

13. Open/Close Juice, Milk Bottle

14. Open/Close Peanut Butter

15. Pour Juice Bottle

16. Pour Liquid Soap

17. Pour Milk

18. Pour Wine

19. Prick

20. Put Salt

21. Put Sugar

22. Put Tea Bag

23. Read Letter

24. Receive Coin

25. Scoop Spoon

26. Scratch, Wash Sponge

27. Sprinkle

28. Squeeze Paper

29. Squeeze Sponge

30. Stir

31. Take Letter from Envelope

32. Tear Paper

33. Toast Wine

34. Unfold Glasses

35. Use Flash

36. Write

Figure 4.3: The 36 actions in our dataset of SoloFinger gestures cover diverse real-world objects and grasps.

Since our approach only focuses on handheld objects, we removed a subset of the actions from this dataset that did not involve grasping an object (performing a high-five; shaking hands; pressing the buttons of a calculator; closing liquid soap). We also combined actions with very similar finger motions and grasp types opening/closing juice and milk bottle; opening/closing peanut butter; and scratching/washing a sponge). We used the video data provided along with the dataset to inform these decisions. As shown in Fig. 4.3, after removal and consolidation, we are left with 36 different hand-object actions (95,788 frames). To use terminology consistent with our second dataset, we use the term "trial" to refer to the sequence of data recorded while one participant performs one action.

4.2.2 SoloFinger Dataset

Thus far, no studies have reported detailed hand data for single-finger movements on diverse grasp types. We therefore recorded a new dataset with hand movement data from study participants who performed SoloFinger gestures while grasping objects. Our focus here is to systematically investigate single-finger movements on a wide variety of objects. However, to evaluate gesture recognition using a more sophisticated model with multiple gesture trials and variations, we collected another dataset as described in Section 4.5.1.

4.2.2.1 SoloFinger Gestures

We centered this first study on three most basic SoloFinger gestures. These comprise the primitive finger movements of *Flexion* and *Extension*, as commonly used in the field of biomechanics, and *Tap* as a discrete motion. These 3 primitive movements are illustrated in Fig. 4.2 (b, top row). For a baseline comparison, we also collected one trial of static *Hold* for every case in which the object was held in a static pose and no gesture performed. Before starting the experiment, we demonstrated these 3 gestures on a cylindrical prop (which was not part of the set of objects used in the study) to familiarize participants with SoloFinger gestures.

4.2.2.2 Actions

We used the same 36 hand actions as in the Hand-Object Actions dataset (as shown in Fig. 4.3). We asked the participants to perform gestures while holding the object in a static pose.

4.2.2.3 Participants

We recruited 15 participants. Not all trials were recorded from 2 participants due to technical reasons. Therefore, all subsequent analysis uses data from the remaining 13 right-handed participants (6 female) aged from 21 to 30 (median=27) from different professional

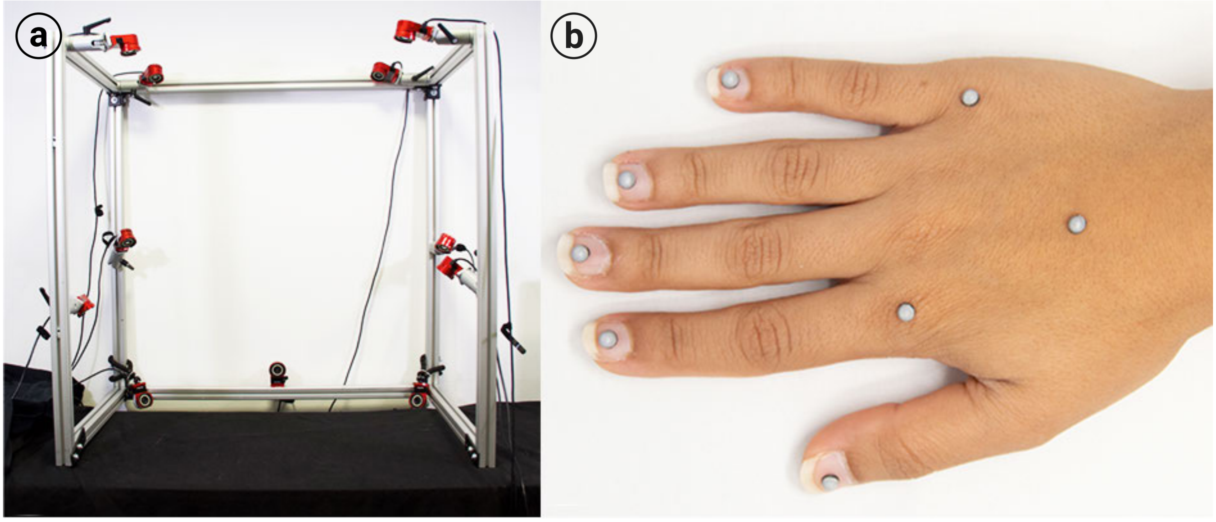


Figure 4.4: (a) Study setup using an OptiTrack motion capture system consisting of 11 infrared cameras. (b) Retroreflective markers placed on the hand to track finger movement.

backgrounds (engineering, law, literature). Participants received a compensation of €20 for their participation. Before collecting data, we manually measured the hand dimensions of participants following the BigHand2.2M approach [119], which involves measuring the distance between different finger joints. We found, on average, distances from the wrist to the tips of: *thumb* - 121mm (SD=8mm), *index* - 141mm (8mm), *middle* - 151mm (9mm), *ring* - 144mm (9mm), *pinky* - 120mm (10mm).

4.2.2.4 Apparatus

We used the OptiTrack™ motion-tracking system with 11 cameras running at a 60Hz refresh rate to capture finger movements. We attached 8 facial reflective markers (4 mm diameter): one on each finger tip and on the wrist, and two on the MCP joint (where the finger connects to the hand) of the index and pinky fingers to help with the manual labeling. The setup is shown in Figure 4.4. To ensure each marker is consistently labeled with a unique ID, we manually annotated markers during post-processing in the OptiTrack's Motive software [120]. In total this results in 879,908 frames of data that we use in our analysis.

4.2.2.5 Task and Procedure

The participant held an object in the dominant hand and used the same hand to perform a SoloFinger gesture on the object with a given finger. We asked participants to perform the gesture in such a way that it felt *comfortable* to them, and not in an exaggerated manner. For each action, the participants had to perform all gestures using all fingers. If they could not perform the gesture due to the odds of dropping the object, they notified the experimenter, who marked this pair of gesture-action as impossible to perform. For possible gestures, they rated each trial for ease on a 5-point Likert scale. We randomized the order of action and counterbalanced the order of gesture and finger used to perform the gesture. For each participant, the experiment took approximately 3 hours and was conducted in two sessions of 1.5 hours each. The full dataset containing recorded data for possible gestures from 13 participants with 5,530 trials is made available to the research community (see Section 4 for the link).

4.2.3 Data Preprocessing

We solely consider the fingertip position to assess finger movements and define the wrist position as the origin of the coordinate system. We apply a median filter on the realigned finger coordinates to reduce the noise. We analyze the hand data using overlapping windows of 1-second duration with a one-frame shift. We chose one second based on the observation that most movements were completed during this interval.

On each window, and for each finger, we create a minimal 3D bounding box to attenuate the jitter in the signal. The bounding box covers the fingertip's 3D positions during the frames that constitute this window. We calculate the diagonal of the bounding box over each window as an estimated measurement of the longest straight line the finger has moved during this window. In the following, we refer to these diagonals as the *movements* of the fingers.

4.3 Concept Validation

The two datasets described above provide data of hand-object interaction, with and without gestures. In the following, we compare them to validate our concept by assessing whether extensive single-finger movements are unlikely to happen during everyday actions. Subsequently, we evaluate the feasibility of performing SoloFinger gestures while grasping objects and derive a set of fingers we recommend for SoloFinger gestures.

4.3.1 SoloFinger Gestures Are Unlikely to Happen During Everyday Hand-Object Actions

The goal of this analysis is to compare finger motions during SoloFinger gestures with everyday hand-object actions. We introduce a *Peak Score* metric that quantifies how strongly the movement of a single finger deviates from the movement of the other fingers. To calculate this, we take the finger with the maximum movement and calculate the ratio between its movement and the movement of the others:

$$PeakScore = \frac{\max_{f \in F} m_f}{\sum_{f' \in F} m_{f'} - \max_{f \in F} m_f} \tag{4.1}$$

where F is the set of fingers and m_f is the movement value over one window for a finger f. We compute this value for all windows in the datasets. A value of 1 shows that a single finger moved as much as the other fingers combined, whereas a value of 0.25 shows all fingers moved the exact same amount. Hence, a high score signifies that one finger traveled a considerably longer distance than the others.

The peak score allows us to numerically compare recorded actions that involve gestures with actions not involving gestures. This provides insights into whether the SoloFinger hypothesis holds. We depict the average peak scores of both datasets for each action in Figure 4.5. One can notice that peak scores for everyday actions are impressively low compared to actions including SoloFinger gestures. A Mann-Withney U test comparing both datasets yields a highly significant difference ($p < 0.001$ with Cohen's d=0.99). We observe an average peak score of daily hand-object actions of 0.33 (SD=0.07). Note, these actions include different finger movements from precise (like plugging the charging cable into the cellphone) to dynamic motions, such as opening or closing peanut butter, or squeezing paper. In contrast, we observe an average peak score of SoloFinger gestures of 1.70 (SD=1.95). The high standard deviations can be explained by the fact that a gesture happens quickly, hence only raising the peak score for an instant.

In addition, we noticed that the peak score for gestures depends on the grasp type and available surface area for finger movements. For example, the Tap gesture has similar peak scores on actions involving similar grasp types (e.g., `give coin` 1.56 (SD=0.90) and `tear paper` 1.49 (SD=0.70)). In contrast, other actions like `pour wine` have a different grasp type, involving all fingers in contact with the object. This provides stability to do extended gestures, resulting in higher peak scores (e.g., `pour wine` 4.19 (SD=3.55) during Tap). With respect to available surface area, the actions with smaller available surface area have smaller peak scores for Flexion and Extension when compared to Tap. This is because the object allows less room for continuous finger movement (e.g., `open peanut butter` shows a peak score of 2.99 (SD=2.66) for Tap; 1.13 (SD=0.98) and 1.08 (SD=0.71) for Flexion and Extension respectively). Further studies should investigate how other object properties, such as texture and weight affect finger movement..

Overall, these findings suggest that extensive single-finger movements are indeed rare during everyday actions, hence creating an opportunity for simple yet robust microgestures.

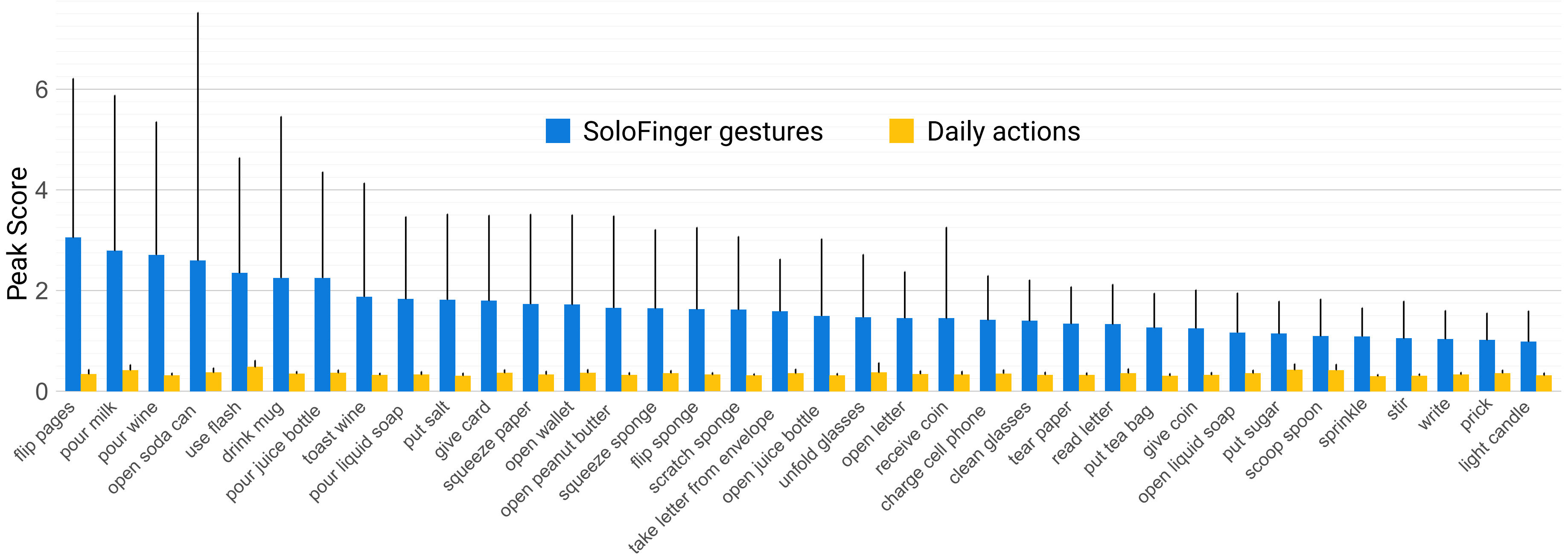


Figure 4.5: Average peak scores for each action present in the two datasets (i.e., with and without SoloFinger gestures). The half error bars depict one standard deviation.

4.3.2 SoloFinger Gestures Are Compatible with Holding Objects

We now focus on the feasibility of performing SoloFinger gestures. While holding an object, the fingers' primary task is to stabilize the object. It needs to be investigated if, despite this primary task, fingers can perform SoloFinger gestures, and if this holds true for diverse object geometries and grasps. To address these questions, we analyzed the gestures participants made in the SoloFinger dataset as well as their subjective ratings.

4.3.2.1 Finger individuation while holding objects

A first prerequisite for performing a SoloFinger gesture is that a single finger can move independently from others while holding the object. The *Individuation Index* (I_{ID}) [121] is a widely used metric from neuroscience that measures the extent to which a finger can move independently from others. If a finger has absolute independence, its I_{ID} is 1.00. Conversely, a value of 0.0 denotes high dependence. We calculate this metric over three windows and retain the maximum value for each trial. Results revealed a high average individuation index(>0.90) [122] for all fingers: thumb = 0.98 (SD = 0.06), index = 0.97 (0.07), middle = 0.96 (0.07), ring = 0.95 (0.07), and pinky = 0.96 (0.09). This denotes the principled possibility of performing single finger movements with all the fingers across a diverse set of actions, comprising diverse object geometries and grasps. To investigate if some actions are more suitable than others, we analysed the I_{ID} for all actions. We found that the I_{ID} is high for all actions, the lowest value being 0.93 (SD = 0.10) for the action `prick`.

Despite this principled feasibility of single-finger movement, it is obvious that not all fingers can be moved while holding an object. Depending on the grasp, some fingers are vital for stabilizing the object; moving those would cause dropping the object. For instance, while picking up a coin, the user cannot gesture with index or middle finger, but might move any other finger. In our data collection, participants attempted to execute each gesture for each action with any of the five fingers. Each combination of finger, action and gesture that a participant considered impossible to perform without dropping the object was labeled as "impossible". Noteworthy, for each action and for all participants, at least three fingers could be used to perform a SoloFinger gesture without dropping the object.

As a further metric investigating if single-finger movements are possible to perform while holding objects, we also measured the average extent of movement of performed gestures. These were: thumb 27.4 mm (SD = 21.1), index 29.1 mm (24.5), middle 36.2 mm (27.5), ring 38.3 mm (30.0), and pinky 45.7 mm (34.2). Overall, these extents indicate a sufficiently large range of motions for reliably performing gestures.

4.3.2.2 Ease of use

For each possible gesture, participants rated on a five-point Likert scale how easy it was for them to perform the gesture. Figure 4.6 shows the normalized ratings aggregated per finger. The results reveal that a vast majority of gestures performed with thumb, index and middle finger are considered easy or very easy to perform. In contrast, approximately half of the gestures performed with ring and pinky fingers were not rated as easy to perform. Mann-Whitney U tests with Bonferroni corrections revealed highly significant differences between all fingers (p¡0.001) except between thumb-middle (p=0.76) and ring-pinky (p=0.76). This indicates users felt more comfortable performing SoloFinger gestures with the thumb, index, and middle fingers.

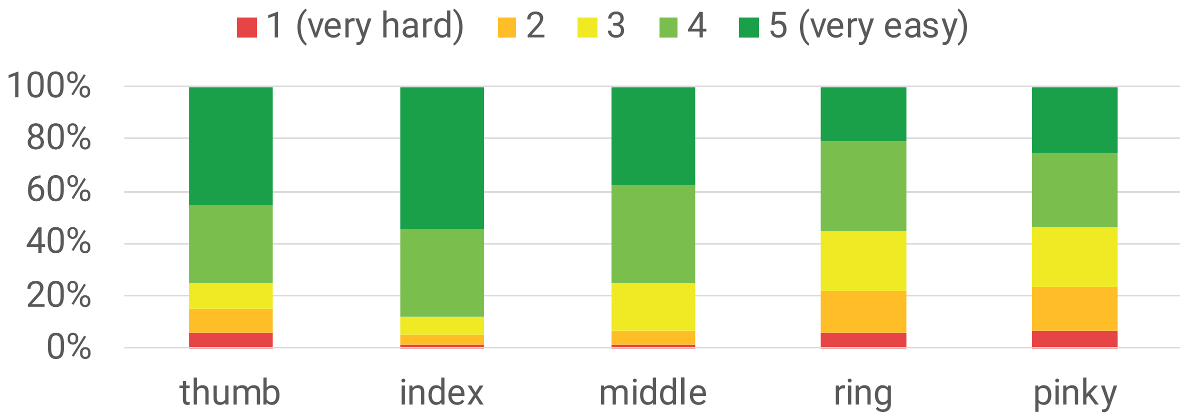


Figure 4.6: Normalized subjective ratings of ease-of-use of SoloFinger gestures (captured for each gesture trial).

A qualitative analysis of the video recordings of gesture trials that have received a low rating revealed that during a considerable number of these actions the ring and/or pinky finger were not in contact with the object, but spread out in mid-air. This comprised actions like `write, sprinkle, clean glasses`, where objects are grasped with thumb, index, and middle fingers mostly. This made it more difficult to perform the gesture, since the participant first had to move the finger in mid-air to bring it onto the object and then move it on the object. Several participants commented about fatigue created by the single-finger movement in mid-air, an effect that is also mentioned in prior work [122].

4.3.2.3 Recommended set of fingers

We conclude that single-finger microgestures can be performed with all fingers. While holding any object comprised in our dataset, a minimum of three fingers is free to move and to perform SoloFinger gestures. This shows that SoloFinger gestures are a viable input method for diverse everyday objects. However, ring and pinky fingers were oftentimes not in contact with the object. According to subjective ratings, the ease of performing gestures with these fingers is significantly reduced. While gesturing with ring and pinky fingers may still function well for select objects, we do not recommend using these fingers in systems that involve a diverse set of objects or grasps. As our goal in this chapter is to investigate gestures that are compatible with versatile objects, we center our following analysis on thumb, index, and middle finger.

4.4 Recognizing SoloFinger Gestures and False Activations

Recognizing gestures from sensor data can be framed as a classification task for a machine-learning model. In our work, we opted for two different approaches: *white-box* and *black-box*. A white-box model is a machine-learning technique that can be easily interpreted by a human. The advantage of such an interpretable and transparent system is that one can understand the decision process of the machine-learning model [123]. We therefore use it as a Design Material [109] in our analysis to derive guidelines for future developers and designers of microgestures. In contrast, black-box models are too complex to allow a straightforward analysis of their learned decision-making. Interpreting such models is an active and open research question [124–126]. Black-box models can, however, use their additional complexity to learn more advanced decision-making processes. This often results in better evaluation performance and is, therefore, usually preferred to achieve state-of-the-art results and for real-world deployment. For example, Wolf et al. [127] found that random forest performed significantly better when comparing the learning-based approach with the threshold-based approach. We train and evaluate such a black-box model to show a proof-of-concept implementation of our concept in Section 4.5.

Thus far, our findings have revealed a principled difference between SoloFinger gestures and finger movement during everyday actions and have confirmed their compatibility with grasping diverse objects. We now set out to assess in more detail the conceptual feasibility of SoloFinger gestures for robust gesture detection. We first present a white-box classification technique using thresholds. This simple model set-up enables a clear human interpretation and understanding of the prediction process. We use it to gain further insights into the use of SoloFinger gestures as well as for the evaluation of false activations in a large, pre-existing dataset of daily hand-object actions. In addition, to validate our concept and overcome the limitations of the white-box classifier, we also present a black-box classifier in Section 4.5.2 that uses a more powerful machine learning model and supports a more complex gesture classification setting.

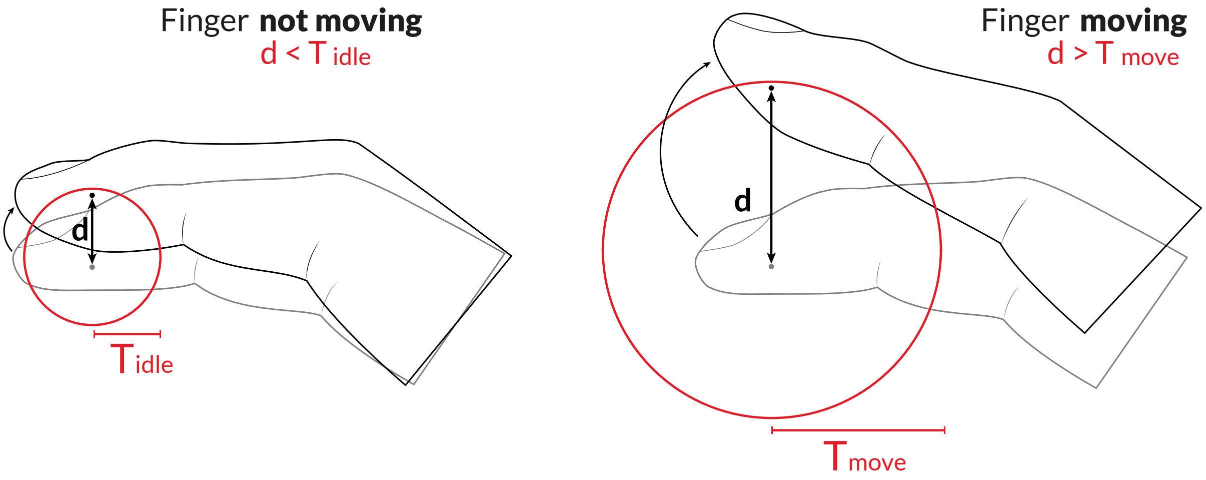


Figure 4.7: Two thresholds for *idle* and *moving* fingers are used to identify single-finger microgestures with the white-box classifier.

4.4.1 White-box Thresholding Classifier: User and Action Independent

As shown in Section 4.3.1, the *Peak Score* of SoloFinger gestures is much higher than in everyday hand-object actions. Given this large difference, we hypothesized that a very simple thresholding technique might be a feasible approach for gesture classification.

To understand if only the extent of finger movement contains a sufficient amount of information, we define two thresholds, one for moving and another for idle fingers, as illustrated in Figure 4.7. A single-finger movement is detected if the movement of a single finger is above the *move* threshold, while all others remain below the *idle* threshold.

4.4.1.1 Train-Test Split and Label Encoding

We used the fingertips' distance as defined in Section 4.2.3. For classification, we first need to identify two thresholds (idle and moving). The recordings of 3 participants were randomly selected and used to find thresholds for their data (Train set). The other 10 participants' recordings were held out and only used for evaluation (Test set). This avoids possible overfitting on evaluation data and allows us to better understand how well this procedure generalizes to unseen data. To compare the classifier's performance to the ground truth for training and evaluation, we obtained labels (gesture or non-gesture class) that were manually annotated. Each trial, consisting of a recording of a gesture or non-gesture for one specific action and finger, counts as one instance. An instance is classified as positive by the threshold model if a single-finger movement occurs in at least one of its windows. For analysis, we built a separate classifier for each gesture (Tap, Flexion or Extension). We, therefore, obtain three separate binary classifiers that check for specific gesture vs. no-gesture.

4.4.1.2 Threshold Optimization

During data collection, we observed that the extent of single-finger movement varies depending on the finger used and the context, i.e., how the user is grasping the handheld object. This suggests one should define tailored thresholds for each finger and adapt these thresholds to each action. Creating individual thresholds for each action, however, is a challenging problem to solve, requiring tracking actions during user interaction, to update thresholds on-the-fly. Hence, for this conceptual evaluation, we aim for a simpler solution and define a consistent set of thresholds for all users and actions. Considering these observations, we aim at tuning 2 thresholds for our feasible set of fingers, i.e., thumb, index, middle fingers and their primitive movement, resulting in a set of $2 \times 3 \times 3$ thresholds.

We run *Bayesian Optimization* to optimize the thresholds on the training data from 3 participants [128]. We have two goals for this optimization process: 1) A gesture recognition system should only predict a gesture if the user really performed one (precision), and 2) The system should also detect as many of the user's intentional gestures as possible (recall) and not miss them. The F_1 score combines both of these goals as the harmonic mean of precision and recall. We, therefore, selected the thresholds that optimized the F_1 score on the training data. As hyperparameters for the search space for both thresholds, we used $\chi_{idle} = [0, 20]$ and $\chi_{moving} = [10, 80]$ mm. Table 4.1 contains the optimized threshold values. We applied the optimized thresholds across thumb, index, and middle fingers to classify individual gestures in the Train and Test sets. Note that we use the same thresholds for all actions and all participants in the following analysis.

Table 4.1: Optimized threshold values (in mm).

Gesture	**Thumb**		**Index**		**Middle**	
	Idle	**Move**	**Idle**	**Move**	**Idle**	**Move**
Tap	16.94	22.45	12.73	21.99	7.66	15.59
Flexion	8.94	20.90	9.64	19.05	6.97	15.49
Extension	10.96	21.48	10.05	16.32	9.35	13.00

4.4.2 Evaluation of Gesture Recognition

Table 4.2 shows precision, recall, and F1 scores for the three primitive gestures, for both Train and Test datasets. On the Test dataset, a 100% precision and a recall of 93% is achieved for Tap. Flexion and Extension achieved a 100% precision and recall of 79% and 83%, respectively.

A video analysis of gesture trials with low recall revealed two main reasons for misclassification, linked to the simple distance thresholding scheme. First, several small objects offer limited surface area, resulting in smaller gestures, some of which were too small to trigger the movement threshold. For example, the `flash spray` head provides a tiny surface for fingers to slide on, resulting in an average recall of only 65% for Flexion and Extension; in contrast, Tap achieved a recall of 96% on the same surface. Second, a few actions include fingers packed closely, thereby limiting individual finger movement. Notably, `drink mug` and `pour milk` involve wrapping the fingers around a confined space, which leads to finger movement smaller than the threshold. A third source of lower recall was one action (`receive coin`) in which the object was held in the palm without any finger contact. Here, fingers were not constrained by the object, and thus idling fingers moved more extensively in mid-air, in turn violating the idle threshold.

Table 4.2: Classification performance of the white-box classifier.

	Train (3p)			**Test (10p)**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Tap	1.00	0.96	0.98	1.00	0.93	0.96
Flexion	1.00	0.82	0.90	1.00	0.79	0.88
Extension	0.99	0.92	0.95	1.00	0.83	0.90

4.4.3 False Activation During Daily Hand-Object Actions

For an empirical evaluation of false activation, we used the pre-existing Daily Hand-Object Actions dataset [116], which extensively covers a wide range of grasps and actions. For human interpretability, we used the white-box classification approach with idle and moving thresholds. Note, when optimizing these thresholds in Section 4.4.1.2, the Daily Hand-Object Actions dataset was held out. The simplicity of the white-box classifier allows us to verify on a trial-by-trial basis where the SoloFinger concept does *not* hold, i.e., under what circumstances an everyday action is misclassified as a SoloFinger gesture. We used our three sets of gesture thresholds and evaluated the whole dataset successively for each set. As the dataset does not contain any SoloFinger gesture, any detected gesture must be considered a false activation. We flagged a trial with false activation if it was triggered by any of the three gestures.

We found that false activation happened in 51 out of 933 trials. Figure 4.8 shows false activation scores per action. The results show that for 23 (out of 36) actions, there were no false activations. Most false activations relate to only five actions. On further analysis, we found actions with most false activations possess two main properties: the object being heavily deformable (paper) or very small (cell phone charger, match stick used to light candle). In both cases, idle fingers do not stabilize and violate the idle thresholds, because they either move along with the deformable object or move in mid-air because they are not in contact with the small object. Interestingly, the `Pour Wine` action triggered a relatively

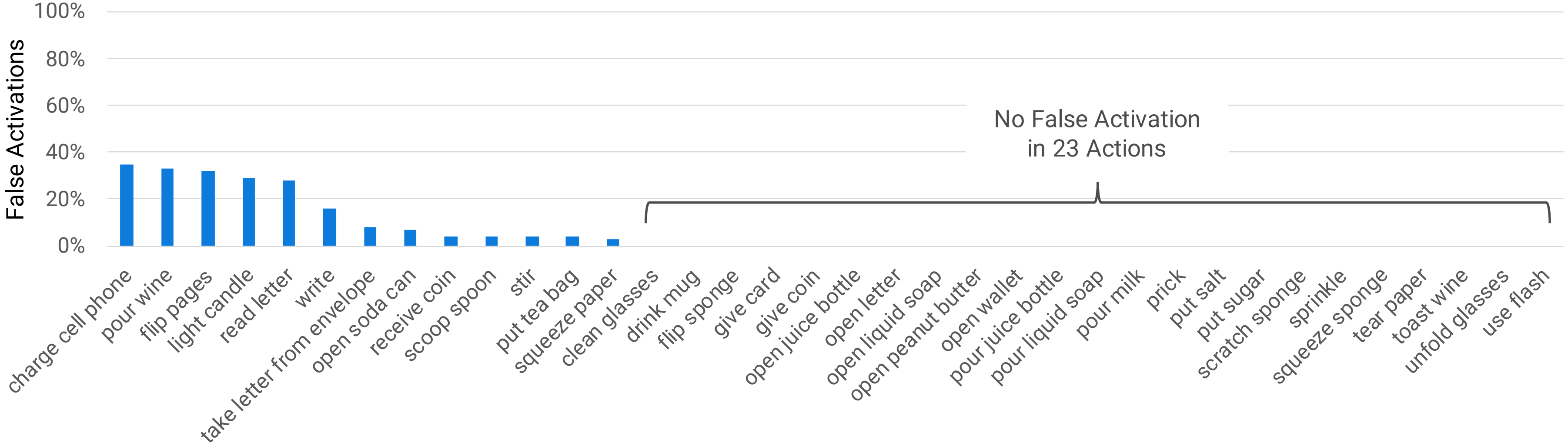


Figure 4.8: Occurrence of trials with false activations in the large dataset with daily hand-object actions.

high number of false activations, despite neither a deformable nor a small object involved. When analyzing the video recordings of these trials, we noticed that four participants used a specific way of holding the bottle with thumb and index finger only, while the middle finger was suspended in air, violating the idle threshold. This was not the case with the other subjects, and as a result no false activation was triggered during their trials.

Overall, these results are encouraging and demonstrate that SoloFinger gestures, even with a very simple classification scheme, lead to little false activation during a wide range of everyday actions. They are particularly robust in cases of everyday actions that include rigid objects and involve three or more fingers in contact with the object. Most false activations related to a few specific actions. Our findings suggest that gestures performed on small objects can be more robustly classified if information about finger-object contact is available. Then, the classifier could be modified to only consider fingers while they are on the object. Classification of gestures made on deformable objects could be improved with information about the position of fingertips on the object, rather than in 3D space. We will show in the next section that even without additional sensor data, classification results can be further improved by adding more feature information beyond the simple thresholds.

4.5 Proof-of-Concept with Commodity Hardware

Our initial study confirmed the principled suitability of SoloFinger microgestures as a robust means for gestural input during diverse everyday actions. We now demonstrate a complete end-to-end recognition system with multiple trials recorded for each class. It is based on commodity hardware – a virtual reality glove – for tracking finger movements and uses a random forest classifier.

4.5.1 VR Glove Dataset

4.5.1.1 SoloFinger Gestures

In addition to the primitive finger movements investigated above (Tap, Flexion, and Extension), we added four more gesture variations: Swipe Left, Swipe Right, Zigzag, and Circle. The gestures are shown in Figure 4.2 (b). Each can be performed with the thumb, index or middle finger, creating a total of 21 interaction options (7 gesture variations × 3 fingers). It is worth mentioning that this is not an exclusive list, and many more variations can be created using the SoloFinger concept. In addition to gesture trials, we also recorded trials while holding the object in a static pose and while performing actions with the object.

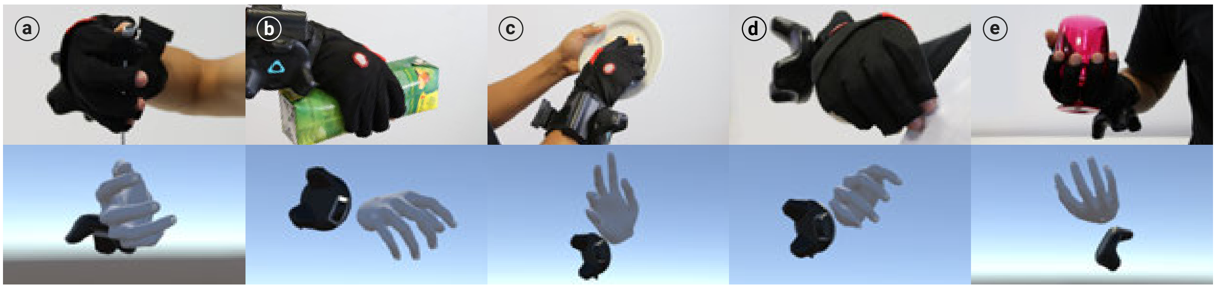


Figure 4.9: Proof-of-concept system using VR glove hardware supporting five frequently used grasps. (a) Charge Cell Phone, (b) Pour Juice Bottle, (c) Scratch Sponge, (d) Take Letter from Envelope, and (e) Toast Wine. The screenshots show the Unity hand model.

4.5.1.2 Actions

To keep the study duration feasible while recording multiple gesture trials for a learning-based classifier, we selected a subset comprising actions corresponding to the five most frequently used grasps, informed by prior work [129]. (The actions are shown in Fig. 4.9). These actions vary considerably in their duration to complete the activity (longer grasp time), involve various motions, and possess different object geometry and rigidity.

4.5.1.3 Apparatus

We use the Noitom® Hi5 VR Glove to track finger movements [130]. The glove provides quaternions for each joint. To capture data in a similar format as the other datasets, we attached a cubical *Unity Game Object* at the fingertips and wrist on the provided hand model.

4.5.1.4 Participants

We recruited 8 right-handed participants (4 female) aged from 22 to 26 (median = 24), including two participants from the previous experiment. We used the same technique as described in our first study (see Section 4.2.2) to measure participants hand sizes. We found, on average, distances from the wrist to the tip of: thumb - 111mm (SD = 12mm), index - 134mm (16mm), middle - 139mm (19mm), ring - 130mm (23mm), pinky - 112mm (14mm).

4.5.1.5 Task and Procedure

We divided the data collection into two parts: 1) collect hand-object action data without gestures, and 2) record SoloFinger gestures. We asked half of our participants to first collect the action data and then record gesture data after a gap of approximately five days, and vice-versa with the remaining participants. For each action, the participants performed all 7 SoloFinger gestures with every possible finger of their dominant hand, except the ring finger. We recorded [8 (#participants) × (5 (#action) × 4 (#finger) × 7 (#gesture) + 5 (#hand-object action) + 5 (#static hold) - 42 (#impossible gestures))] × 10 (#trials) = 8640 trials. Informed by the empirical findings described in Section 4.3.2, we removed the pinky gestures and did not consider data with ring and pinky for further classification, resulting in 5,840 trials. The labeled VR Glove Data along with the precise finger movement data captured by OptiTrack are available at the link mentioned in Section 4 to facilitate future research in this area.

4.5.2 Black-box Classifier

The white-box classifier provided insights about the properties of SoloFinger gestures at a general level with user and action independent thresholds. The real-world deployment, however, may provide an option either to fine-tune the thresholds in a calibration process [8], or to leverage the complex decision boundaries used by the black-box classifiers to support multiple gestures. Here, we present such a system to support multiclass classification.

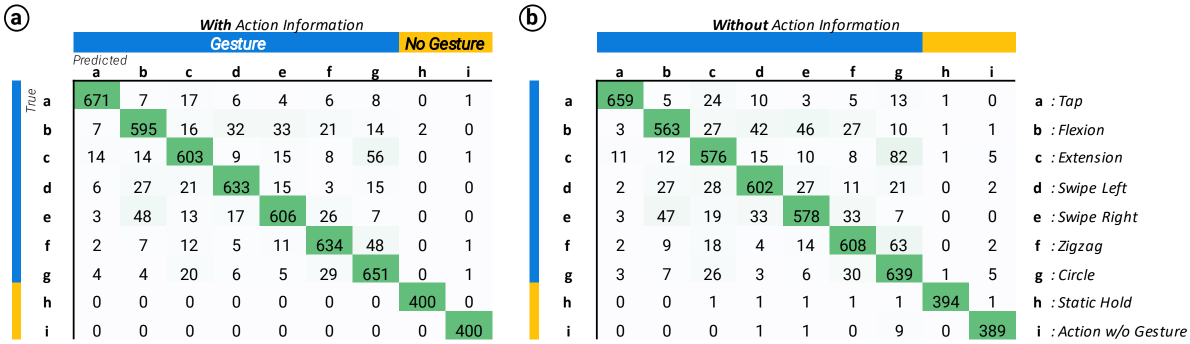
4.5.2.1 Data Preprocessing

Similar to our previous data preprocessing strategy, we used the raw 3D coordinates of the Unity Game Object and defined the wrist position as center. Subsequently, we applied a median filter on the realigned coordinates.

4.5.2.2 Feature Representation and Classification

TsFresh [131] is used to obtain a feature representation for each instance, i.e. for the sensor recording sequence of each individual trial. Subsequently, we fed the feature representation in a random forest classifier provided by Sci-kit Learn [106]. This pipeline is not specific to our system and was used in prior work [62]. We did not optimize the hyper-parameters of the classifier and used the default settings with n-estimators = 500. Due to the personalized patterns involved in some gestures, such as Zigzag or Circle, that have a high degree of variation among users, we opted for user-dependent models. In a real setting, pre-trained models for individuals can be easily saved and restored to avoid the burden of per-session training. From our initial analysis, we learned that finger movements vary on different objects. Therefore, we performed evaluation in two conditions - training with and without action information. Thus, the classification task was to classify individual trials of all 9 classes (7 gestures + 1 static hold + 1 action without gesture) without action information for every participant. For the condition with action information, we trained models separately for all five actions.

We evaluate classification using the leave-one-trial-out 10-fold cross validation technique for each participant (9 Train trials of each class, 1 Test trial - 10 permutations). Note that


a
With Action Information
Gesture
No Gesture
Predicted
True
b
Without Action Information

	a	b	c	d	e	f	g	h	i
a	671	7	17	6	4	6	8	0	1
b	7	595	16	32	33	21	14	2	0
c	14	14	603	9	15	8	56	0	1
d	6	27	21	633	15	3	15	0	0
e	3	48	13	17	606	26	7	0	0
f	2	7	12	5	11	634	48	0	1
g	4	4	20	6	5	29	651	0	1
h	0	0	0	0	0	0	0	400	0
i	0	0	0	0	0	0	0	0	400

	a	b	c	d	e	f	g	h	i
a	659	5	24	10	3	5	13	1	0
b	3	563	27	42	46	27	10	1	1
c	11	12	576	15	10	8	82	1	5
d	2	27	28	602	27	11	21	0	2
e	3	47	19	33	578	33	7	0	0
f	2	9	18	4	14	608	63	0	2
g	3	7	26	3	6	30	639	1	5
h	0	0	1	1	1	1	1	394	1
i	0	0	0	1	1	0	9	0	389

a : Tap
b : Flexion
c : Extension
d : Swipe Left
e : Swipe Right
f : Zigzag
g : Circle
h : Static Hold
i : Action w/o Gesture

Figure 4.10: Confusion Matrices with and without action information.

the imbalance in the trial count is because we combine the results across all actions which include impossible gestures on a few fingers.

4.5.3 Results

4.5.3.1 With Action Information

Figure 4.10 (a) shows the result of the gesture classification for a setting in which the action is known. Note that no false activations were triggered. The average accuracy is 89%. Across all actions, the Tap gesture achieved the highest accuracy of 93%, followed by Circle with 90%. The lowest accuracy was 83% for Flexion and 84% for Extension. We assume this is related to the VR glove's wiring, which runs across the back of the fingers and might have restricted bending movements. In contrast to results from the white-box classifier, where fingers violated the idle threshold on extremely deformable objects (paper), the black-box classification did not trigger any false activations for the action that involved paper (`take letter from envelope`).

Note that this model requires prior knowledge about the action the user is performing. In many applications, this information is readily available. For instance, a microgestural input system devised for a specific object (such as a smart surgical tool, an augmented drilling machine, or a smart pen) can readily use an object-specific model. Otherwise, activity recognition [132] could be integrated to identify the ongoing action.

4.5.3.2 Without Action Information

For comparison, we report results for the more demanding case in which the action is not known. Here, the average classification accuracy is 86%. Here again Tap achieved the highest accuracy of 92%. The lowest accuracy of 84% was Circle, which sometimes gets misclassified with Zigzag, particularly when performed with the middle finger. False activations were triggered only in 2.12% of total action and static hold trials, with a total of 17 (out of 800) trials (see Figure 4.10 (b)).

4.6 Discussion and Limitations

Our findings demonstrate that SoloFinger gestures provide a robust scheme for microgestural input on the object itself, creating a low number of false activations during many everyday hand-object actions. Here we reflect on the strength and limitations of the proposed.

4.6.1 Sensing Technology to Implement SoloFinger

SoloFinger is not restricted to the hardware we use in our proof-of-concept implementation and can be implemented with various sensing technologies. A sensor should provide information about the distance that a user's fingertips move, at approximately a 2–3 mm resolution. Of note, the sensor does not necessarily need to provide information about finger-object contact. However, an important requirement is that it function while the fingertip is in contact with an object. While this prevents us from using established computer vision techniques, which tend to suffer from occlusion generated by the object [133], recent advances show promising results for hand-object interaction [133–135]. Objects equipped with high-resolution touch sensors [63, 136–138] are also promising to further deploy SoloFinger gestures, given the precise temporal and spatial contact information provided. Furthermore, other technical approaches such as magnetic [76, 139], electro-magnetic [75], radar-based [140], and IMU-based approaches [89] can be promising avenues for realizing SoloFinger gestures.

4.6.2 Gesture Classification

Our results revealed that false activations are primarily caused by deformable and small objects, such as paper or a match stick. Our current scheme considers data from fingers no matter whether in contact with the object or not. In our white-box analysis, we found that in-air fingers tend to make considerably larger involuntary movements while another finger is gesturing, due to the lack of stabilizing object contact. Hence in-air fingers more frequently violated the idle threshold and therefore led to lower recall. Future implementations could detect finger contact using a dedicated sensor or approximate it based on grasp or action type, and then only consider fingers that are in contact with the object for classification. We showed that adding more features and a more advanced classifier as in our black-box implementation can further help increase classification performance and robustness. Furthermore, using an ensemble of classifiers may further improve the accuracy of gesture detection: a binary classifier, as described in the white-box classification, forms a first layer to identify the gesture and non-gesture class; this is followed by a second layer of multi-class classification.

4.6.3 Investigating More Objects and Specialized Actions

While our analysis did demonstrate that SoloFinger gestures could be very robust in reducing false activations during daily hand-object interactions using diverse objects, there remain additional cases to be investigated. First, our study did not include objects that comprise mechanical interface elements, such as buttons or sliders, or touchscreens. We assume that using those might involve extensive single-finger movements similar to SoloFinger gestures. In these cases, we recommend that the designer should carefully understand the regular finger movement on such objects and then select SoloFinger gestures accordingly to avoid conflicts. Second, we observed that highly deformable objects can result in increased false activation. Our datasets do not contain information about the touched location on an object; hence it does not allow us to differentiate between finger movements on an object and fingers that remain at the same object location but move in 3D space while the object itself is deforming. With a sensing technology that provides on-object touch location, we anticipate that the SoloFinger concept could still work for most deformable objects. Furthermore, the effect of specialized dexterous actions need to be studied. Actions such as playing a musical instrument, sculpting or performing a surgery might possibly involve more pronounced single-finger movements.

4.7 Conclusion

This chapter presented *SoloFinger*, a novel concept to identify and design single-finger microgestures that are robust while grasping everyday objects. The results from our data-driven analysis confirmed the insight that fingers tend to be either static or moving concurrently while holding and manipulating a wide range of objects. This opens up a space for rapid, easy and elegant microgestures performed by a single finger on the object itself and resilient to false activations. Our simple white-box classifier achieved an average precision of 100% and recall of 88%, with only 51 false activations among 933 action trials of an unseen dataset. Of note, no false activation was triggered in 23 actions out of 36 actions.

We ultimately presented a proof-of-concept with commodity hardware and a black-box classifer that can detect 7 types of SoloFinger microgestures and hand actions with an accuracy of 89%. When the action is known, no false activations occurred in the collected dataset of around 800 everyday actions, whereas a small number (2.12%) of trials involved false activation in the more demanding case when a single classifier is used for all actions.

Inspired by our peers [141, 142], we also provide our dataset to the community to further advance the understanding of dexterity of single-finger movements while grasping objects and leverage this dexterity to design quick and seamless gestures that can be integrated with everyday actions. Despite the fact that covering the full hand with a large number of sensors can help detect gestures in situations where hands hold objects, in the real world, it is desirable to be able to have a minimal set of sensors that can perform gesture classification at a higher accuracy. The following Chapter 5 introduces an efficient method for detecting system microgestures in both free and busy hands with minimal hand instrumentation.

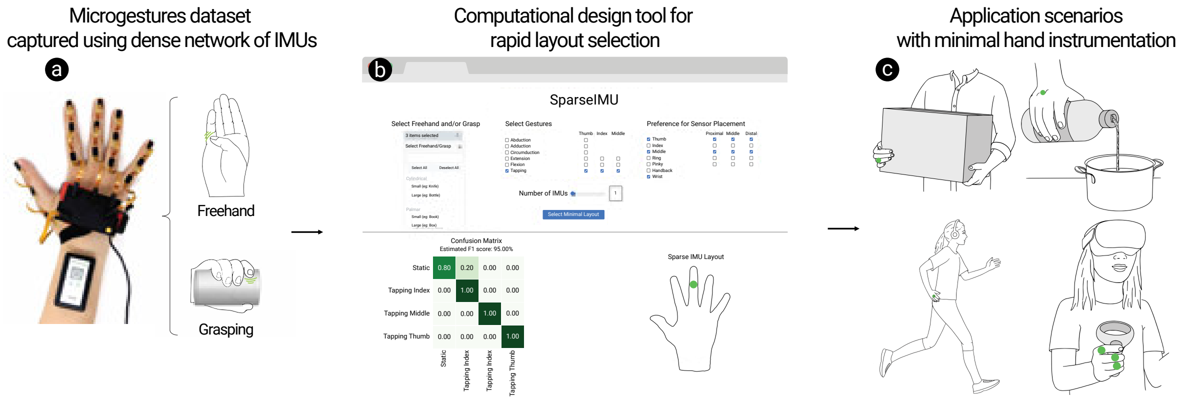
Chapter 5

Computational method for designing sparse sensor layouts to detect fine-grained microgestures

In Chapters 3 and 4, we have primarily focused on designing gestural input techniques to overcome the physical constraints caused by holding an object in hand. Our focus now shifts to implementing these gestures. Implementing always-available input recognition systems requires consideration of several form-factor requirements and technical issues. Noteworthy, it becomes even more challenging if the system should detect microgestures in both freehand and busy-hand scenarios, which involve multiple grasp types. An ideal device should be minimally invasive, low cost, and require low power. As mentioned in Chapter 2, inertial measurement unit (IMU) sensing can overcome occlusion issues, detect finger movements, and reconstruct the full hand when multiple sensors are attached to different joints and work synchronously. Due to IMUs' sensitivity to movement, a minimal form factor device, such as a smart ring worn on an appropriate finger segment, can be used to sense finger movements.

Sparse sensing is a well-known principle in signal processing, has demonstrated promising results in various problems to achieve a higher classification rate by optimally positioning sensors [96]. We employ this principle together with the IMU sensing and gesture classification. Specifically, this chapter moves beyond the conventional strategy of manually trying out multiple locations to determine the optimal sensor locations and present *SparseIMU*[1] – a computational design approach to assist interaction designers and engineers in developing

[1] This chapter's contents are based on an article published at TOCHI, which I led as the first author [22]. I formulated the problem space, came up with the rapid method using feature importance, designed the dataset collection studies, collected the dataset with the dense IMU network, led the data analysis, and designed the GUI tool. Additionally, I started and led the collaboration with the Charité University Hospital and TU Berlin researchers.



SparseIMU

Select Freehand and/or Grasp

3 items selected

Select Freehand/Grasp

Select All Deselect All

Cylindrical

Small (eg: Knife)

Large (eg: Bottle)

Palmar

Small (eg: Book)

Large (eg: Box)

Select Gestures

	Thumb	Index	Middle
☐ Abduction	☐		
☐ Adduction	☐		
☐ Circumduction	☐		
☐ Extension	☐	☐	☐
☐ Flexion	☐	☐	☐
☑ Tapping	☑	☑	☑

Number of IMUs 1

Select Minimal Layout

Preference for Sensor Placement

	Proximal	Middle	Distal
☑ Thumb	☑	☑	☑
☐ Index	☐	☐	☐
☑ Middle	☑	☑	☑
☐ Ring	☐	☐	☐
☐ Pinky	☐	☐	☐
☐ Handback			
☑ Wrist			

Confusion Matrix
Estimated F1 score: 95.00%

	Static	Tapping Index	Tapping Index	Tapping Thumb
Static	0.80	0.20	0.00	0.00
Tapping Index	0.00	1.00	0.00	0.00
Tapping Middle	0.00	0.00	1.00	0.00
Tapping Thumb	0.00	0.00	0.00	1.00

Figure 5.1: We present a data-driven method for designing effective microgesture recognition systems that only require a sparse set of IMUs. (a) The method builds on an extensive microgestures dataset that includes Freehand and Grasping conditions, collected using a customized dense IMU setup. (b) A design tool helps designers to rapidly select sparse IMU layouts for a desired set of gestures and optional constraints. (c) It informs effective sensing solutions with minimal instrumentation for a broad variety of applications.

gesture recognition systems. Importantly, our method supports both *freehand* and *grasping microgestures* for creating sensor layouts with minimal hand instrumentation (**RQ III:** What sensor locations on the hand provide effective recognition with minimal instrumentation?). In addition, we contribute a GUI-based design tool that enables designers to specify high-level requirements and designer-specified constraints (e.g., desired gestures and grasps, locations on the hand and fingers that remain un-instrumented, number of IMUs to be deployed). Based on these inputs, the tool automatically selects an optimal sparse IMU layout matching the given preferences as shown in Figure 5.1-b. The tool also predicts the expected performance of gesture classification, including a confusion matrix. This allows the designer to assess the expected quality of a solution and to rapidly explore design alternatives in a well-informed manner. To the best of our knowledge, our computational approach and design tool are the first to facilitate the rapid, iterative design of sparse IMU-based microgesture solutions.

The presented data-driven approach is based on our collection of an extensive microgestures dataset, captured with a customized hardware setup containing 17 synchronized IMUs placed all over the dominant hand. It comprises of 18 gestures and three non-gesture states performed with an empty hand as well as on 12 objects that cover all the six grasp types from Schlesinger's taxonomy [15], collected from 12 participants. Our dataset comprises fully annotated dense IMU data. This allowed us in evaluating the entire combinatorial space for freehand and grasping microgestures (393K IMU layouts).

To investigate the potential of making conscious design choices when selecting a specific sparse IMU layout, we performed a series of empirical analyses looking into effects on recognition performance. Chiefly we have made the following observations: i) Sparse layouts with a very low number of IMUs achieve high recognition rates of 90% F1 score and above, ii) the choice of finger segment for IMU placement can be crucial, and iii) IMUs placed on a non-gesturing finger can be utilized to detect gestures from another finger. These findings reveal insights that uncover the great potential of sparse IMU layouts in gesture detection.

The collected microgestures dataset additionally serves as the building block for deriving a fast method to select sparse layouts. We employ a variant of a well-known metric from Machine Learning (ML), *Feature Importance*, to rapidly select optimized sparse layouts. We validate our SparseIMU approach with the classification results from the entire combinatorial space; the results demonstrate our method's efficacy. While generating results based on the entire combinatorial space is prohibitively time-consuming for a practical design task, our method generates results within minutes on a commodity laptop. Consequently, our approach can be used to enable rapid design iterations. We demonstrate the benefits of the SparseIMU approach using four exemplary application cases. Finally, our user evaluation shows congruence in the tool's predictions and live gesture recognition. These show how the tool enables designers and engineers to rapidly determine optimal sparse IMU layouts, identify trade-offs, and fine-tune designs. Together, our rich microgestures dataset and computational design tool enable a rapid iterative design process in which designers can create, explore and modify custom sensor layouts in a well-informed manner.

We release our fully annotated microgestures dataset captured using 17 IMUs placed on the hand with microgestures and hand manipulations with freehand and while holding 12 objects, performed by 12 participants - overall, it consists of 13,860 trials (3.4M frames). In addition, we also share the computational tool code at: *https://hci.cs.uni-saarland.de/projects/sparseimu/*

In the following, we first describe the Microgestures dataset in Section 5.1. Then, we present results from a series of analysis in Section 5.2. We introduce our faster method for layout selection and the corresponding evaluations in Section 5.3. Next, the GUI-based design tool and its evaluation with the entire combinatorial classification results are presented in Section 5.4. It additionally covers the state-of-the-art techniques of gesture sensing, recognition, and computational design tools. Section 5.5 describes application scenarios from diverse and representative domains to illustrate how designers and engineers can benefit from our design tool, which is followed by the evaluation of the tool's output and live gesture recognition in Section 5.6. Discussion and Limitations are presented in Section 5.7. Finally, Section 5.8 describes this chapter's conclusion.

5.1 Microgestures Dataset

Researchers in the computer vision community have contributed various datasets comprising hand-object manipulations [116, 143, 144]. Yet, these do *not* include explicit finger gestures. Our dataset is the first attempt to collect hands-free and busy interaction along with finger microgestures. We use a dense network of 17 IMUs to capture high-dimensional sensor data with nearly full degrees of freedom (DOFs) of the hand/finger space. This is different from prior work wherein a single sensor has been shifted to different locations in different trials for finding the optimal placement [89]. Our high-dimensional data enables employing novel algorithmic approaches to uncover hidden phenomena; some of them are mentioned in the following sections. Overall, our dataset focuses on finger gestures – performed by different fingers – on objects with diverse grasp types, as well as with free hands. It also comprises hand-object manipulations with different intents, such as holding an object, using it as suggested by its primary purpose (e.g., writing with a pen), and handling it in an unscripted manner (e.g., fiddling). Although the dataset is intended to analyze microgestures, it can serve other purposes in future research, including enriching our understanding of finger movements during hand-object interaction, creating synthetic data, or pre-training neural networks.

5.1.1 Dense IMU Setup

Instead of utilizing commercially available gloves or marker-based solutions [14, 145], we performed the data collection with a customized hand sensor system that preserves the cutaneous properties of the hands, the sense of touch, and does not suffer from occlusion. The sensor system is shown in Figure 5.2. It offers an unobtrusive setup of 17 synchronized IMUs [146, 147] that provide detailed information about the full articulation of a human hand. It includes 9DOF inertial sensors with 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer (MPU9259, InvenSense Inc., CA, USA) with a footprint of 3×3 mm, deployed on all three segments of all five fingers using a medical-grade skin-friendly adhesive tape (Helvi Mogritz).

The finger IMUs are mounted on flexible sensor strips and connected to a base unit attached at the hand's back, which includes an additional IMU. A customized fixture with a thin velcro belt is used to fasten the base unit on the hand, and the data is sent to the computer through a USB connection. We also attached a wireless IMU (RehaGait, Hasomed GmbH, Germany) on the distal forearm, to include data comparison from existing consumer devices like smartwatches or fitness trackers, resulting in a total of 17 IMUs. All IMUs are precisely time-synchronized, and the data is captured at a framerate of 100 Hz. We refer to Salchow-Hömmen et al. [146] for full details on formal hardware validation, which found that sensor readings are accurate enough to infer fingertip positions with errors

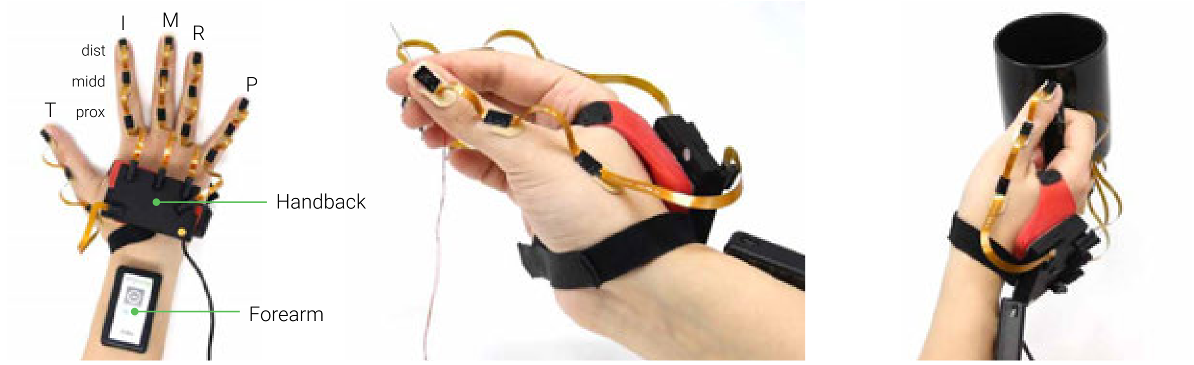


Figure 5.2: Hardware setup with 17 synchronized IMUs placed all over the dominant hand. It preserves cutaneous properties and allows unobtrusive interaction with complex object geometries. The left image labels describe the spatial notation of each IMU used in our analysis.

< 2 cm. For the use of the raw IMU data, the hardware does not require any calibration, making it particularly practical and feasible for studies. However, we integrated an initial pose with the hand flat on the table and the straight thumb abducted at a known angle for a few seconds at the beginning of each subject's recording, in order to boost the dataset's versatility in light of potential future uses where a baseline or calibration pose might be desired. We also note that the framerate of our dense setup of 17 IMUs is in line with that of Xu et al.'s [148] recent work, which suggests that 100 Hz is sufficient for hand gestures' classification. Furthermore, prior studies have found that even the quick movements of the fingers are slower than 10 Hz [149, 150].

5.1.2 Objects Representing Grasp Variations

We collected data in Freehand and while Grasping an object conditions. For the latter, we selected a set of objects that are representative of real-world tasks. Specifically, we chose objects labeled in the *VLOG Dataset* [151] which is based on internet video logs of everyday activities. To ensure we have representatives for each type of grasp, we categorized the objects based on Schlesinger's Grasp Taxonomy [15]; this has been widely employed by prior works [12, 20, 152, 153]. For each grasp type, we focused on non-deformable objects with two size variations Small (S) and Large (L). The VLOG Dataset does not contain objects that correspond to *Small Tip* and *Spherical* grasps, which is presumably a result of not all grasp types being equally well-represented in everyday life [129]. Therefore, we added two additional objects, a Needle and Pestle, to obtain an exhaustive list of objects covering all grasp types [20]. The complete set of 12 objects and their corresponding grasp type is shown in Figure 5.3.

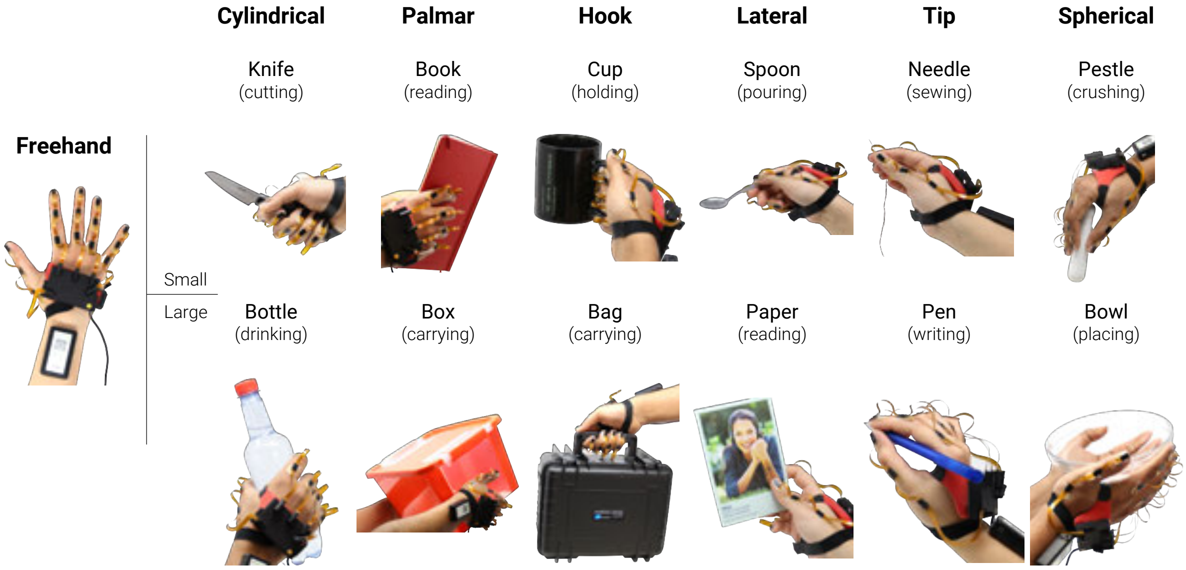


Figure 5.3: Using a dense network of 17 IMUs placed on the hand, the microgestures dataset was collected for Freehand and while Grasping 12 objects covering each of the six grasp types with two variations.

5.1.3 Gesture Set and Non-Gesture States

For Freehand and Grasping conditions, we collected finger movements while performing microgestures and non-gesturing states. For the microgestures, we focused on conscious subtle finger movements that do not require altering the grasp. We selected six primitive finger movements based on bio-mechanical characteristics [154, 155], shown in Figure 5.4: *Tap*, *Flexion*, *Extension*, *Abduction*, *Adduction*, and *Circumduction*. For consistency of gestures across different fingers, we use the Ring finger as the reference to define Abduction (away from the Ring finger) and vice-versa for Adduction gestures. Furthermore, the swipe gesture was recorded with the participant's finger motion from one extreme until it reached the opposite extreme. Following Ashbrook's definition of micro-interactions [37], we further limited our set to gestures with a short duration (4 seconds or less). Moreover, we centered our data collection on single-finger gestures because they promise to increase robustness [21]. In terms of gestural input, these movements translate to both - continuous and discrete gestures through directional sliding and tapping.

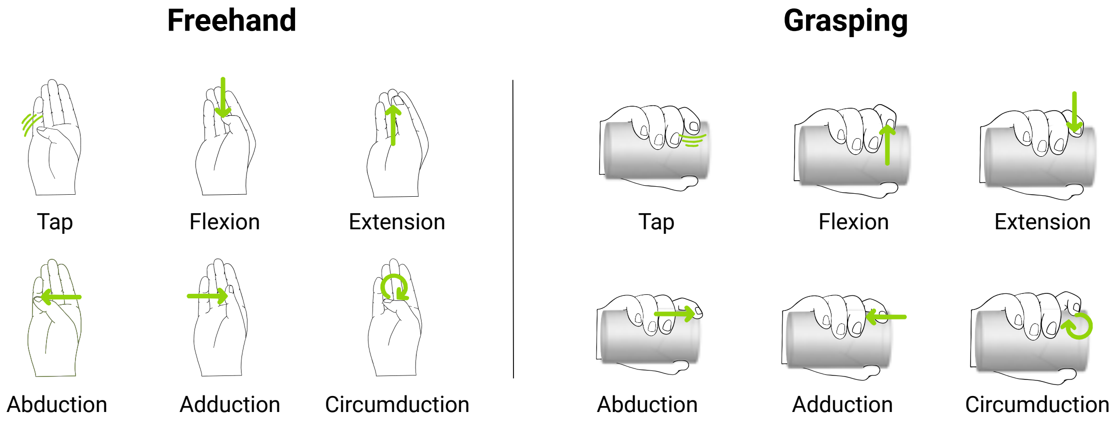


Figure 5.4: The Dataset includes six gestures performed with three fingers - Tap, Flexion, Extension, Abduction, Adduction and Circumduction - resulting in a total of 18 gestures. Additionally, data was recorded for three non-gesture classes: Static hold (just holding the object), performing Primary action while holding the object, and an Unscripted action where the user was free to perform any custom movements.

The collected non-gesture states include a variety of finger movements that users perform consciously or unconsciously during conventional hand/object interaction. For instance, free hand movements while talking, adjusting the grip, turning the object for visual inspection, manipulating the object, or fiddling. For capturing non-gesture conditions, we recorded *Static hold*, *Primary action* (e.g., writing with a pen, drinking with a glass), and *Unscripted actions* (e.g., adjusting grip, fiddling). The participants were given no explicit instructions while the data for Unscripted action was recorded.

Since moving a finger while holding an object risks dropping the object, we empirically verified which fingers can be moved while holding objects. To consolidate our choice of finger movements, we conducted a pilot study. Two interaction design experts independently recorded their response on a 7-point Likert scale (1: impossible to perform and leads to dropping the object; 7: very intuitive and easy to perform). This resulted in a total of 360 gestures: 6 (gestures) × 5 (fingers) × 12 (objects) inspected by each expert. Of 720 Likert scale readings, 42 gestures received a rating of 1 by both the experts and these were marked as impossible. Consequently, we focus on the Thumb, Index, and Middle fingers as our main gesture fingers; a choice which is in-line with prior works [4, 21].

5.1.4 Participants

We recruited 12 participants (6M, 6F, mean age: 26.1; SD: 3.4) with different professional backgrounds, including computer graphics researcher, firefighter, and kindergarten teacher. Ten were right-handed, and two reported themselves as ambi-dexterous. We measured their hand size from the Wrist to each finger's tip and found an average length to Thumb's tip: 137mm (SD:8mm), Index: 181mm (SD:12mm), Middle: 192mm (SD:12mm), Ring: 181mm (SD:10mm), and Pinky: 157mm (SD:9mm). For context, the average hand length (middle finger's tip to the wrist crease) is 193 mm and 180 mm for males and females, respectively [156]. Participation to our data collection was voluntary while adhering to the institution's COVID-19 rules and regulations, and each participant received a compensation of 30 Euros.

5.1.5 Task and Procedure

Before starting the data collection, we demonstrated the gestures on an abstract cylindrical object that was not used any further. Once the participants got familiarized with the gestures, we attached the hardware to their dominant hand, and they performed the initial pose by placing the hand on the table. For the Grasping condition, we asked the participants to perform gestures on the object (while maintaining the grasp), and use the palm as the surface for the Freehand condition. Of note, the same hand was used for holding the object and for gesturing. Furthermore, the directional orientation was kept constant across each participant. They performed all the gestures while sitting on a chair, except for Box and Bag, wherein we systematically added variation in posture and orientation for each participant by asking them to perform the gestures while standing and facing perpendicularly. We counterbalanced the two conditions (Freehand and Grasping) and further counterbalanced the order of objects (grasp variations). Once the Freehand or the Grasp variation was selected, we presented the gestures with the specific finger name and non-gesture states in a randomized order. We recorded five trials for each gesture. To collect data from non-gesture states without interruption, we recorded one long sequence of around 30 seconds and split it into five trials. The dataset collection took approximately 3 hours per participant with breaks in-between to avoid fatigue. The sessions were also video recorded. Using a custom MATLAB application, the experimenter manually annotated the trials during data collection with the participants orally communicating the start and stop of the gesture. The labels include information about the freehand or specific grasp variation, gestures along with the instructed finger, and the three non-gesture states. Overall, our dataset contains a total of 13,860 trials (1,155 trials × 12 participants) with 18 different gesture and three non-gesture states performed on 12 Grasp variations and with Freehand.

5.2 Dataset Analysis to Understand IMU Placement

The usage of IMUs in HCI has been explored for gestural input; the most common approach is to place a single IMU on the gesturing finger [88, 89, 157–159]. However, very little is known about the relationship between the precise position of IMU(s) and its effect on classification performance. To understand the multitude of factors affecting the overall classification performance, we sought to systematically investigate different perspectives, including the quantity of IMUs, variation between different finger segments, alternative IMU placement location to simultaneously achieve higher recognition and usability, lastly, evaluate the feasibility of a user-independent recognition model. An in-depth understanding would not only enable taking full advantage of the IMU sensing capabilities and fine-tuning IMU placement to achieve the maximum performance for a given set of gestures, but also uncover hidden patterns to identify optimal designs of gesture sensing devices.

This section first describes our classification pipeline and a series of empirical analyses, which offers new insights into the design of sparse IMU layouts for hand microgesture recognition.

5.2.1 Feature Extraction and Classifier Selection

Aiming to understand the underlying factors affecting performance rate due to IMUs' location, we started off by creating a classification pipeline. Given the size of our search space has the large number of 393K layouts, we created a gesture detection pipeline with two essential requirements: scalable and rapid train-test time.

5.2.1.1 Feature Extraction

From a given trial and for each of the 9 axes of an IMU, we extract six statistical features: maximum, mean, median, minimum, standard deviation, and variance. In total, the number of features from all 17 IMUs × 9 axes × 6 features amounts to 918. To compile this list of features, we drew inspiration from the automatic feature extraction library, TsFresh [131], which has shown promising results in prior work on gesture and activity recognition [62, 160, 161]. Due to multiple sensors and reduced computational load, we used the minimum configuration of the library's functionalities. To further minimize the effect of different trial lengths, we removed the sum and length features. Due to the lower sampling rate of our 17-IMUs setup as compared to single-sensor approaches [57], we did not extract features from the frequency domain. However, we note that our released dataset will allow the research community to feed more features of TsFresh into the neural network [161], take advantage of a single feature, such as derivatives as input into the neural network [89], or further perform feature engineering for input in non-neural-network or neural-network classifiers to improvise the recognition rate based on the optimal location. In Section 5.2.1.4, we show the correlation of our selected features and a different set of features from related work to show the correlation in the ranking of layouts.

5.2.1.2 Method

We selected 10 random participants as training set and the remaining two as test set (80:20 split) and created grasp-independent models, i.e., the class labels do not include any grasp information. We also performed a leave-one-person-out analysis in Section 5.2.5. For our multi-class classification, we used 19 classes: (3 fingers × 6 gestures) + 1 Static hold. Different IMU layouts may contain different amounts of IMUs (from 1–17); therefore, to compare different state-of-the-art classifiers and estimate the classification time required for the full combinatorial classification, we evaluated randomly selected 100 layouts for a given IMU count of 1–17, totaling 1,435 layouts. Note, for count = 1, 16, and 17, the total possible layouts are slower than 100.

5.2.1.3 Classifier Selection

We fed our extracted features into multiple commonly used classifiers to evaluate their recognition rate and training time. Specifically, we used scikit-learn's implementation of Support Vector Classification (SVC), Logistic Regression (LR), k-nearest neighbors (KNN), Random Forest (RF) with max_depth = 30; and PyTorch implementation for Neural Network (NN) with 4 fully connected layers of decreasing hidden layer size (n = 1024, 512, 256, ReLU activation) and a final softmax activated classification layer. Only NN models were trained on a GPU machine and others on a 40-core CPU. We used the default parameters for all the classifiers to perform trial-by-trial basis classification. As a performance metric, we used the macro average of the F1 score because it considers both precision and recall.

5.2.1.4 Results

As shown in Figure 5.5, the F1 score and training time largely depend on the choice of classifiers. Since we wanted to use the same classifier for multiple settings in the following

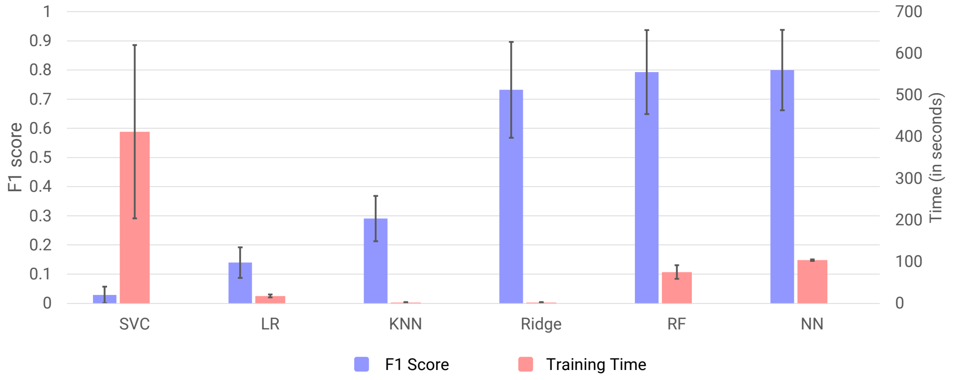


Figure 5.5: Comparison between average F1 score obtained by different classifiers and their training time for 1,435 IMU layouts. The error bars depict one standard deviation.

analyses, as well as the later-described computational design tool (see Section 5.4) – we opted for Random Forest. This classifier achieves an average F1 score close to the highest one obtained by Neural Network while having a lower training time than Neural Network. Furthermore, RF models can be easily computed on a consumer-grade CPU machine. In-line with findings from prior work [127], our results show that Random Forest Classifier has superior performance than KNN.

As shown above, our released dataset allows generating results with various classification models techniques. Through our analysis, we found that, while different models may yield different accuracy levels, the order of performance of individual layouts is very similar. Specifically, to understand our results' dependence on a particular classifier, we used F1 scores of all layouts with sensor count = 1 from the top-performing classifiers, namely KNN, Ridge, RF, and NN. Following that, we sorted the results alphabetically by IMU labels. Then, using a pairwise Spearman correlation (as used by Guzdial et al. [162] for comparing ranked lists), we obtained a correlation of 0.919, 0.975, and 0.919 with $p<0.001$ for RF vs. KNN, NN, and Ridge, respectively.

In addition, we conducted a similar analysis to understand the change in the ranking of IMUs for different sets of features. We selected five features (maximum, minimum, mean, skewness, and kurtosis) used in the existing literature on IMU sensing [88] and trained 17 models with RF. Subsequently, similar to the analysis comparing different classifiers, we calculated the Spearman correlation on the F1 score of alphabetically-sorted IMU's list from both feature sets. Our results show a high correlation of 0.995 with $p<0.001$ between the layout ranking produced by 2 different set of features, indicating that while selecting other features may result in a different F1 score, the order of IMUs remains very similar.

5.2.2 Identifying Sparse Layouts for a Given IMU Count

The large count of IMUs offers the possibility of creating vast layout combinations. However, not every count and layout may produce a similar recognition performance. Therefore, an important aspect that we examined was identifying the best performing sparse layout for a given number of IMUs. This analysis provides three major insights: Firstly, it allows us to understand how the recognition performance varies with the number of IMUs. Secondly, it gives insights into the interval in which F1 scores fall for any given number of IMUs. Lastly, the results inform the optimal IMU placement location with a fixed budget of sensors [96]. Of note, we use the term *IMU Count* to refer to any given amount of IMUs from 1–17.

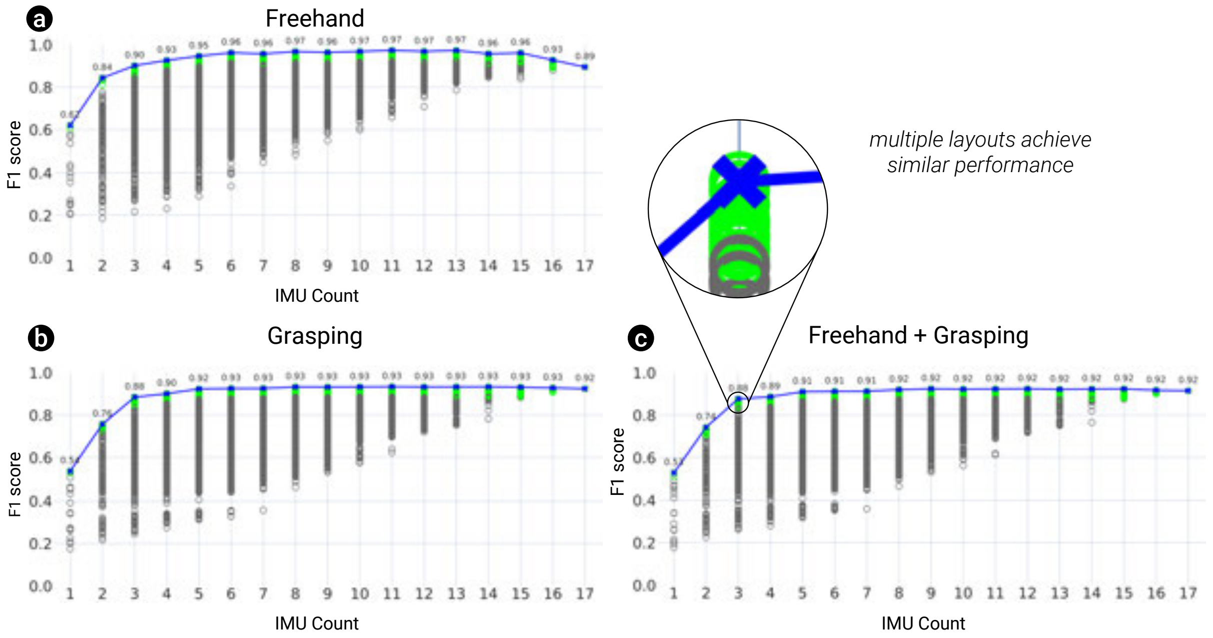


Figure 5.6: Full Combinatorial Results: Each circle represents the F1 score for each of the 393K models classifying 19 classes in Freehand, Grasping, and Both Combined (Freehand+Grasping) conditions. The blue shows the maximum F1 score, and the green depicts the top 5% layouts in a particular IMU count.

5.2.2.1 Method

To explore the full combinatorial space, we trained models with all possible layouts from 1 to 17 IMUs on our initial train-test split as described in Section 5.2.1.2. Moreover, to systematically understand the variation in performance for both types of microgestures, we performed this analysis for three conditions: Freehand, Grasping, and Both Combined. This totals to 3 × $(2^{17} - 1)$ = 393,213 models. For each model, we performed multi-class classification with 19 classes: (3 fingers × 6 gestures) + 1 Static hold. Note, Grasping and Both Combined conditions utilized grasp-independent models; therefore, we did not encode grasp information in the class labels. In Section 5.2.6, we compare our results with grasp-dependent models.

5.2.2.2 Results

Figure 5.6 plots the F1 score on the test set from each 393K models trained in all three conditions (Freehand, Grasping, Both Combined), organized by the count of IMUs present in the model. We now discuss each condition in turn:

1. Freehand microgestures: The results provide a complete overview of the large performance difference that depend on the IMU count and, for a given IMU count, on the specific location of IMUs comprised in a model. As shown in Figure 5.6-a,

the highest F1 score for count = 1 is $0.62_{\text{(M-midd)}}$. Adding a second IMU increases the F1 score to $0.84_{\text{(T-midd, M-dist)}}$; the F1 score further increases to $0.90_{\text{(T-midd, I-prox, M-midd)}}$ and $0.93_{\text{(T-midd, I-prox, M-dist, R-prox)}}$ with 3 and 4 IMUs, respectively. On the contrary, the lowest F1 score for count = 1 was $0.2_{\text{(Forearm)}}$, and for count = 2 was $0.19_{\text{(R-prox, Forearm)}}$. Amongst all models, the maximum F1 score of $0.97_{\text{(T-prox, I-dist, I-prox, M-dist, M-midd, R-midd, P-midd, Forearm)}}$ is achieved with count = 8. It should also be noted that a F1 score of 0.90 can be achieved with as little as 3 IMUs, and henceforth only a maximum increase of 4% occurs with the addition of more IMUs. The F1 score drops to 0.89 when all 17 IMUs are included. To further investigate this drop, we trained 100 classifiers with random states from 0-99 for count = 17. We only change the seed values for this investigation, while training classifiers for other analyses have a constant seed value with default parameters to allow reproducible results. Out of 100 models, 4 models achieved the maximum F1 score of 0.96, which is close to the maximum F1 score of 0.97 achieved by some other higher counts. Overall, 93 out of 100 models achieved an F1 score of greater or equal to 0.90, and only 7 models have an F1 score in the range of 0.88 (lowest) and 0.89. This explains the reason for the drop we observed at count = 17.

2. Grasping microgestures: Here, our classification setting is more challenging than Freehand microgestures due to the inclusion of all 12 Grasp variations. This results in a slight drop in overall performance (see Figure 5.6-b). For count = 1, the highest F1 score was $0.54_{\text{(I-midd)}}$. Adding an additional IMU (count = 2) gradually increased the performance to $0.72_{\text{(I-prox, M-midd)}}$, for count = 3 to $0.88_{\text{(T-dist, I-prox, M-prox)}}$, and for count = 4 to $0.90_{\text{(T-dist, I-midd, I-prox, M-prox)}}$. Similar to Freehand , the IMU located on the forearm achieved the lowest F1 score of 0.17 for count = 1. Across all models, the maximum F1 Score of $0.93_{\text{(T-dist, I-dist, I-prox, M-dist, M-prox, Handback)}}$ is first achieved at count = 6. Note, the general pattern of variation in the maximum and minimum F1 score is similar to the Freehand condition, and an F1 score of 90% can be observed with a small number of IMUs (count = 4). Afterwards, the maximum increment in F1 score is only 3%.

3. Both Combined microgestures: As shown in Figure 5.6-c, we observed a similar overall trend when gestures in Freehand and all Grasp variations were classified together. The maximum performance achieved with one IMU was $0.53_{\text{(I-midd)}}$. Adding more IMUs resulted in an increase of F1 score to $0.74_{\text{(I-prox, M-midd)}}$, $0.88_{\text{(T-dist, I-prox, M-prox)}}$ and $0.89_{\text{(T-dist, I-prox, M-midd, M-prox)}}$ for IMU count = 2, 3 and 4 respectively. Conversely, the minimum F1 score for counts = 1, 2, 3 and 4 is 0.18

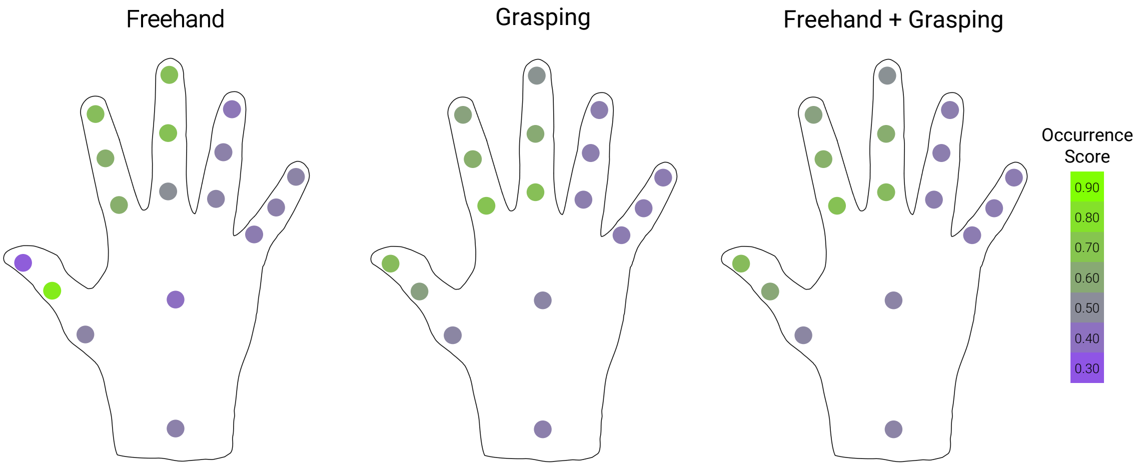


Figure 5.7: Occurrence Score of each IMU in the top 5% layouts from count 1 to 17. Across all IMUs, we observed a minimum score was 0.33 and maximum of 0.84.

(Forearm), 0.23 (P-dist, P-midd), 0.26 (P-dist, P-prox, Forearm), 0.28 (P-dist, P-midd, P-prox, Forearm) respectively. The min and max difference of the F1 score within each IMU count shows a similar pattern as the other two conditions. Across all counts, the maximum F1 score of 0.92 (T-dist, T-midd, I-dist, I-prox, M-dist, M-midd, M-prox, R-dist) is first achieved with count = 8. At count = 5, an F1 score of 91% is obtained, and only a 1% increase is seen with more IMUs.

5.2.2.3 Relevance of each IMU

Multiple layouts may achieve a performance close to the top-most layout in each count as shown in Figure 5.6. To better understand what locations on the hand and finger are more likely to contribute to top-scoring layouts, we analyzed the top 5% best-scoring layouts (marked in green color in Figure 5.6). Specifically, we introduce an *Occurrence Score* metric that quantifies the occurrences of each IMU in the top 5% layouts (see Eq. 5.1). Here, a higher score of an IMU indicates its frequent presence in the top layouts. For a set I of possible IMUs, the Occurrence Score of an IMU i is

$$occ_i = \frac{1}{|I|} \sum_{k=1}^{|I|} \frac{\text{occurrences of IMU } i \text{ in top 5\% layouts with } k \text{ sensors}}{\text{number of top 5\% layouts with } k \text{ sensors}} \tag{5.1}$$

where we calculate the mean of an individual IMU's occurrence over all IMU counts. It is important to note that this is not the overall occurrence in the total space of 393K models but rather how frequently it occurs in the top layouts.

5.2.2.4 Results

We examined the Occurrence Score of each IMU as shown in Figure 5.7 and derived patterns that guide our further analysis. Since the gestures were performed by Thumb, Index, and Middle fingers, the IMUs from these three fingers appear more often in the top 5% layouts in all three conditions (Freehand, Grasping, and Both Combined). Interestingly, the Occurrence Score varies greatly across different segments of the same finger. The comparison between Freehand and Grasping conditions revealed three considerable differences: First, we observe that an IMU placed on the tip of the Thumb (T-dist) has a high Occurrence Score of 0.67 for Grasping microgestures, whereas it is only 0.33 for Freehand microgestures. We assume this is related to the nature of gestures performed on the palm in the Freehand condition, wherein the Thumb stretches out at a larger distance and bends lesser than during Grasping microgestures. In a typical grasp, the Thumb supports the object; hence the distance to reach the surface for performing a Grasping microgesture is relatively smaller. Second, for all fingers except the Thumb, Grasping microgestures tend to favor IMU placement on the proximal segment over the fingertip. In contrast, Freehand microgestures show a clear tendency to favor placement on the fingertip for Index and Middle fingers. Below, we investigate the effect of IMU position on classification performance in more detail.

5.2.2.5 Implications

For all three conditions, we noticed that a higher IMU count does not necessarily translate to higher recognition performance. F1 scores close to the optimal can be achieved already with a fairly small number of IMUs (3 to 6). We observed a large variation in performance depending on where a given number of IMUs is placed on the hand and fingers, which also depends on the microgesture condition as shown in Figure 5.7. These findings highlight the importance of creating a layout by choosing a right number of IMUs, a right combination of fingers, and finger segments for the desired set of grasp and microgestures to achieve optimal recognition accuracy.

5.2.3 Performance of IMU Placement at Segment Level

Having identified that the choice of finger segments for IMU placement can be crucial for obtaining high recognition performance, we now aim to investigate the influence of finger segments on recognition performance more systematically. This also informs the design of minimal form-factor devices that place IMUs only at the optimal segment.

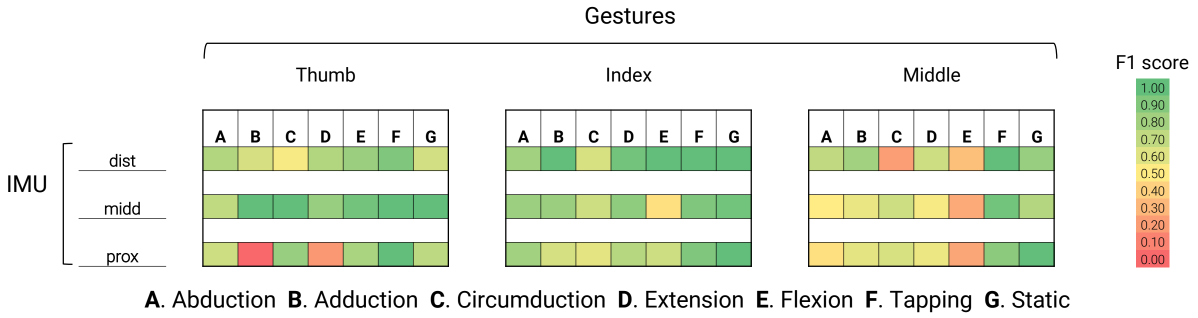


Figure 5.8: F1 score of single IMU models trained for multi-class classification. The classes include six different gesture types possible with each finger (+1 static) for each model during Freehand microgestures. Note that different models were trained with IMU on each segment (distal, middle, proximal) and for different gesturing fingers.

5.2.3.1 Method

We used our initial 80:20 train-test split of the participants' data and evaluated using a single IMU under multiple settings. To reduce any effects caused by different grasp variations, we created grasp-dependent models. Moreover, for a clear understanding of individual fingers and their respective gestures, we performed finger-wise classification, i.e., atmost six gestures and one static hold class per finger. Overall, we trained 17 single-IMU layouts × [(1 Freehand × 3 gesturing fingers) + (9 Grasp variations × 3 gesturing fingers) + (3 Grasp variations × 1 gesturing finger)] = 561 models. For the analysis in this section, we focus on the IMU on gesturing fingers and on three representative grasp variations that have been identified in prior work to each represent a cluster of Grasping microgestures [20]. The detailed results, including IMUs on non-gesturing fingers and all 12 grasp variations will be released with our dataset.

5.2.3.2 Results

As illustrated by Figures 5.8 and 5.9, the F1 score varies greatly across different segments for Freehand as well as Grasping microgestures. In particular, it indicates that for some cases, the F1 score for a gesture may even rise from 0.0 to 1.0 depending on what segment the IMU is placed on the same finger. In the following, we highlight this effect for Freehand as well as Grasping microgestures.

1. Freehand: The kinematics for each finger varies, and the motion required for each gesture is also different. As a result, the F1 score can have a large difference across segments (shown in Figure 5.8). We observed that the optimal segment is different for different fingers. In particular, for Thumb gestures, the middle segment (midd) achieved an average F1 score of 0.93, whereas the other two segments, i.e., distal (dist) and proximal (prox), have a relatively lower score of 0.72 and 0.60, respectively.

The optimal segment for Index gestures is different: here, the prox-segment has an average F1 score of 0.91, while the performance on the other two segments is considerably lower with 0.78 (I-midd) and 0.76 (I-prox). For the Middle gestures, all segments achieved a similar F1 score of 0.60-0.65, the segment choice is still prominent for individual gestures wherein the performance may differ with 20-40% for Adduction, Abduction, and Circumduction. In contrast, the performance difference across segments is lower for the Tap gesture (10–13%). Surprisingly, due to the hand bio-mechanics, the IMU on the Handback can detect Thumb Flexion and Tap with an F1 score of 0.82 and 0.70, respectively. This finding can be beneficial to detect finger gestures in settings where a user might not want to wear any sensor on the finger (e.g., while working in a kitchen or car workshop). We investigate this aspect of recognizing gestures from a non-gesturing finger in more detail in the next section.

2. Grasping: Our results reveal a strong influence of segment choice for Grasping microgestures (see Figure 5.9). Similar to the Freehand condition, we observed a large difference in F1 score across different segments of the same finger. Furthermore, it is noteworthy that there are dissimilarities in the pattern of optimal segment across different grasp variations. This relates to the distinctive finger postures in different grasps, affecting how a finger moves while performing the gesture. In particular, for the Thumb and Index gestures on Cylindrical-S and Spherical-S, the dist segment appeared as the optimal segment in both grasp variations. However, for the Middle finger gestures, the optimal segment is different across all three grasp variations (Cylindrical-S has dist, Lateral-S has mid, and Spherical-S has prox). Moreover, the Index and Middle gestures on Spherical-S have a relatively lower variance across segments, which could be explained by the bigger real estate that affords comparatively larger movements than the other two grasp variations. In general, the substantial difference in the recognition performance at the segment level is due to the intricacies of the grasp variation, finger, and gesture.

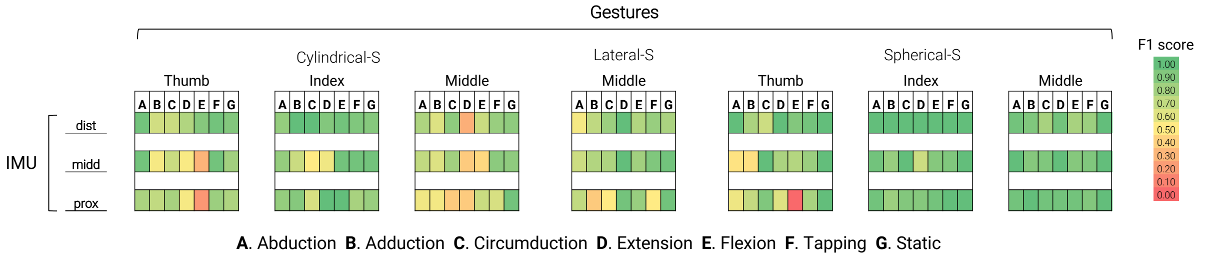


Figure 5.9: F1 score of single IMU models trained for multi-class classification. The classes include 6 different gesture types possible with each finger (+1 static) during three exemplary grasp variations (Grasping microgestures).

5.2.3.3 Implications

Depending on the grasp, finger and type of movement during the gesture, the single-IMU performance across segments greatly varies. This formally validates our initial findings from the full combinatorial classification results: The choice of finger segment for the IMU sensor placement can have a very strong influence on classification performance. However, since these classification results differ based on the subset of grasps and chosen gesture classes, a one-fits-all design solution will likely not lead to best results. Hence, we propose a computational design tool in Section 5.4, which provides layout recommendations based on the user-defined parameters.

5.2.4 Placing IMU on a Non-gesturing Finger

Finger co-activation is a widely known phenomenon in bio-mechanics [121]. Our goal is to leverage finger co-activation and investigate if micro-movements caused in neighboring fingers are sufficient for gesture detection from a non-gesturing finger. This would be beneficial in situations where placement of an IMU on the gesturing finger would hinder the primary activity–e.g., having an IMU on the Index finger may hinder situations like using a knife. In such scenarios, placing the IMU on an alternative location capable of detecting gestures from a neighboring finger would be more desirable.

5.2.4.1 Method

To investigate the possibility of detecting gestures with any single finger, we used our initial 80:20 train-test split and trained five models for each of the three gesturing fingers; each model comprised a total of three IMUs placed on every segment of the respective finger. For a detailed analysis, we performed grasp-dependent and finger-wise classification. This gives a total of 5 fingers w/ IMUs × 3 gesturing fingers = 15 models for Freehand. We trained another 150 models [(5 fingers w/ IMUs × 9 grasp variations × 3 gesturing fingers) + (5 fingers fingers w/ IMUs × 3 grasp variations × 1 gesturing finger]. In each multi-class model, we included all six gestures for an individual finger and the static class - totaling up to seven classes.

5.2.4.2 Results

Figure 5.10 and 5.11 show the F1 score on the test set for Freehand and Grasping when models are trained with IMUs on different fingers. These results indicate the feasibility of detecting gestures from IMUs on the non-gesturing finger:

1. Freehand: We observed the effect of finger co-activation and the feasibility of detecting gestures from IMUs on a non-gesturing finger for all three gesturing fingers (see

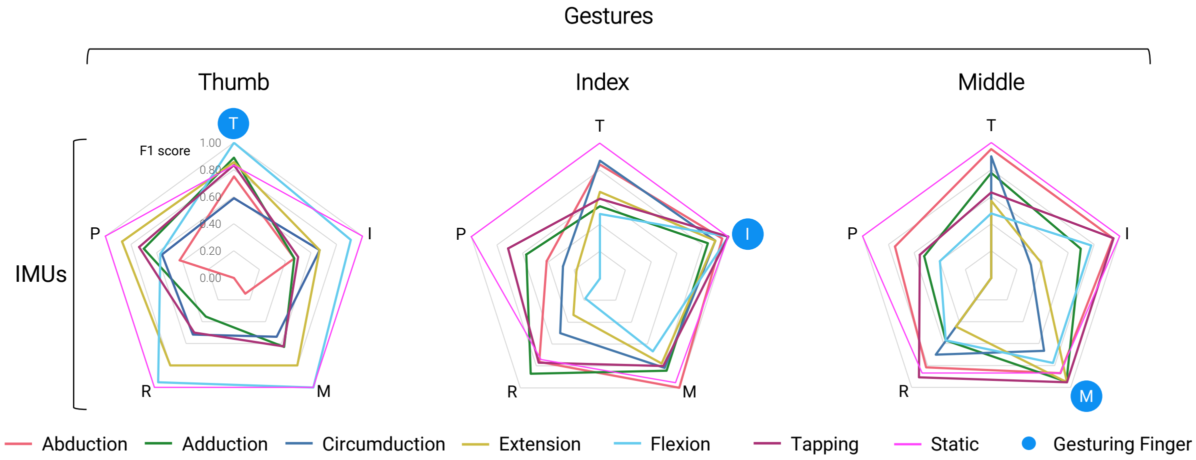


Figure 5.10: F1 score of IMUs placed on gesturing as well as non-gesturing fingers for multi-class classification. The classes include six different gesture types possible with each finger (+1 static) for each model during Freehand microgestures. T, I, M, R, and P refer to the IMUs on Thumb, Index, Middle, Ring, and Pinky finger. The gesturing finger is denoted with a blue circle.

Figure 5.10). Unsurprisingly, placing an IMU on the gesturing finger results in a higher F1 score in most cases. However, it is important to note that depending on the finger and gesture, the IMUs on a non-gesturing finger can even yield a higher F1 score than when placed on the gesturing finger. This is particularly visible with gestures performed by the Middle finger. This observation is in line with findings from prior work that have reported the middle finger to induce higher involuntary movement in adjacent fingers [121, 122]. For Middle Circumduction, for instance, the F1 score on a non-gesturing finger (Thumb) increases by 34% (from 0.67 to 1.00) compared to placing an IMU on the gesturing finger (Middle). This can be explained by the involuntary Thumb movement caused while performing the Middle Circumduction on the palm. Also, Index Adduction achieved a 5% higher F1 score through placing IMUs on a non-gesturing finger (Middle) than gesturing finger. Even though Thumb has the least tendency amongst all the fingers to induce movements in the neighboring fingers, placing an IMU on the non-gesturing finger (Middle or Ring) produces a similar F1 score as that on the gesturing finger (Thumb) for Flexion, Extension and Circumduction. These promising results of placing an IMU on the non-gesturing fingers show the feasibility of detecting gestures beyond the conventional placement strategies.

2. Grasping: As mentioned in prior work, fingers in contact with the object get support, thereby reducing the effect of co-activation [21]. Thus, all Thumb and Index gestures on Cylindrical-S (Knife) achieved the highest performance when the IMUs are placed on the gesturing finger. In spite of that, we observed that the non-gesturing finger can detect Thumb and Index gestures with a drop of only 15–20% from the F1 score

obtained by an IMU on the gesturing finger. While this reduction is considerable, it may be acceptable for some gestures in settings that do not allow for augmenting the gesturing finger with IMUs. Based on the grasp type and gesture, the IMUs on the non-gesturing finger may even achieve a higher performance than the gesturing fingers, e.g., on Spherical-S (Pestle), Thumb Extension and Circumduction achieved a higher F1 score of 0.83 and 0.95, respectively, through IMUs on the non-gesturing finger (Index). In contrast, the IMUs placed on the gesturing finger (Thumb) achieved a comparatively lower score of 0.67 and 0.87. On Cylindrical and Spherical grasps, all fingers are in close contact with object but not all grasp types have the same contact fingers. For example, while holding Lateral-S (Spoon), the Ring and Pinky fingers are suspended in the air, which causes an involuntary movement in the other adjacent non-gesturing finger As a result, the gesturing (Middle) and non-gesturing (Pinky) finger IMUs achieve a similar F1 score for Middle Abduction and can also detect Middle Flexion with an F1 score of 0.80 (0.15 lower from the IMUs on the gesturing finger). Additionally, we observed the possibility of detecting gestures with non-gesturing fingers that are in contact with the object. With these many different factors affecting the performance, it is challenging for a designer to place the sensor at an alternative location intuitively.

Implications When the hands are busy, instrumenting gesturing fingers might not be possible in all cases. For example, while writing, instrumenting fingers involved in gripping the pen might hinder the primary activity. In such scenarios, placing an IMU on neighboring fingers can be efficient. Our findings show that placing IMUs on a non-gesturing finger may enable gesture detection at a comparable or even higher performance rate.

5.2.5 Generalizability of Layouts across Participants

Next, we aim to understand the extent of inter-personal differences in recognition performance. This is a crucial question because there can be inter-personal variations in the way the microgestures are performed. If there is a large difference in classification results across participants, the design tool that we describe in later Section 5.4 would need to account for it while suggesting a sparse layout.

5.2.5.1 Method

A comprehensive Leave-one-person-out (LOPO) evaluation with 12 participants × 393,213 layouts = 4,718,556 models will approximately take 25 days of computation time on our 40-core machine. To circumvent this problem, we first identified the best layout according to the F1 score for a given count of IMUs on our 80:20 participants split from

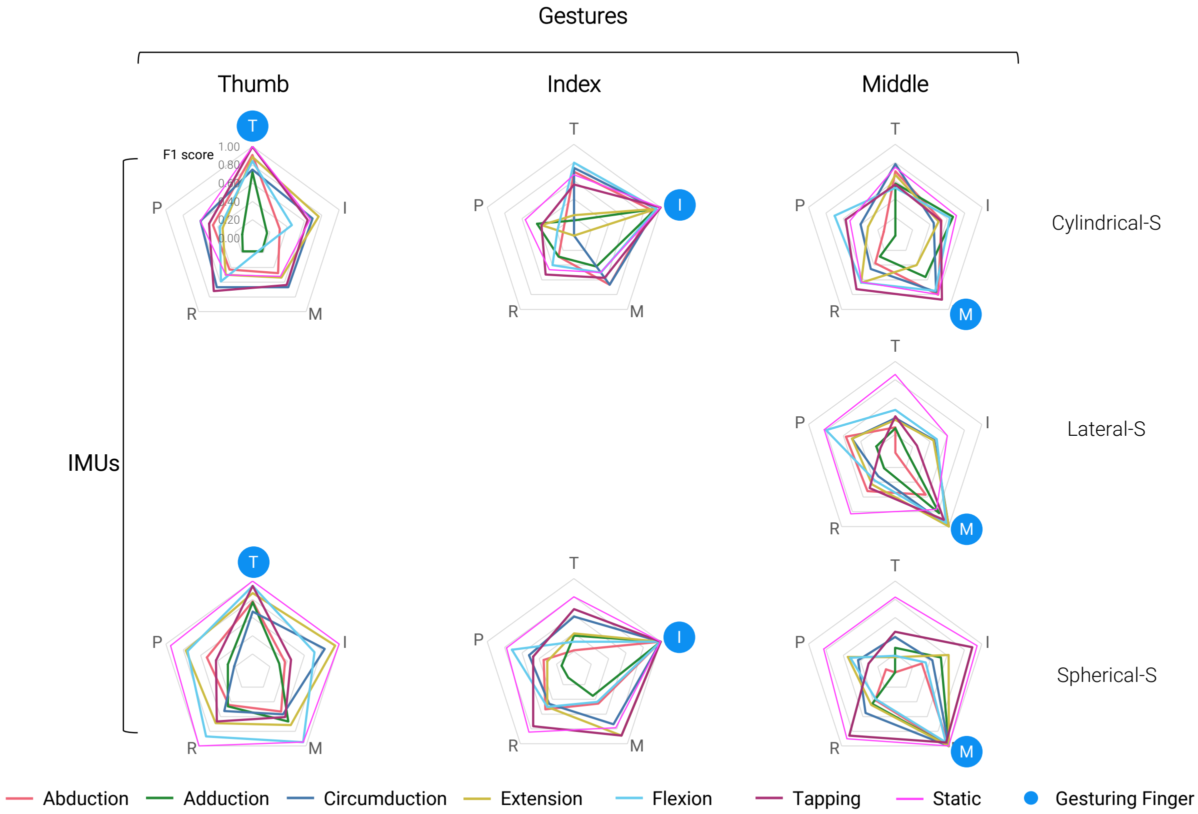


Figure 5.11: F1 score performance of six different gesture types possible with each finger (+1 static) when the IMUs are placed on gesturing as well as non-gesuring fingers for three representative Grasp variations (Grasping microgestures).

the combinatorial results obtained with the combined condition (Freehand+Grasping). Subsequently, we used these best layouts and trained 204 models (12 participants × 17 best layouts for the IMU Counts) for a LOPO evaluation.

5.2.5.2 Results

Figure 5.12 depicts the results of the LOPO evaluation. We observe that the difference in F1 score from our randomly selected 80:20 train-test split and any LOPO model is about ±6%. It is worth noting that most participants achieved higher performance than our randomly chosen participants.

5.2.5.3 Implications

Despite the inter-personal variations in how the gestures are performed, our recognition pipeline still scales well and achieves high recognition performance with user-independent models. We observed only little variation in F1 scores across participants, which demonstrates that model predictions generalize to data from new users.

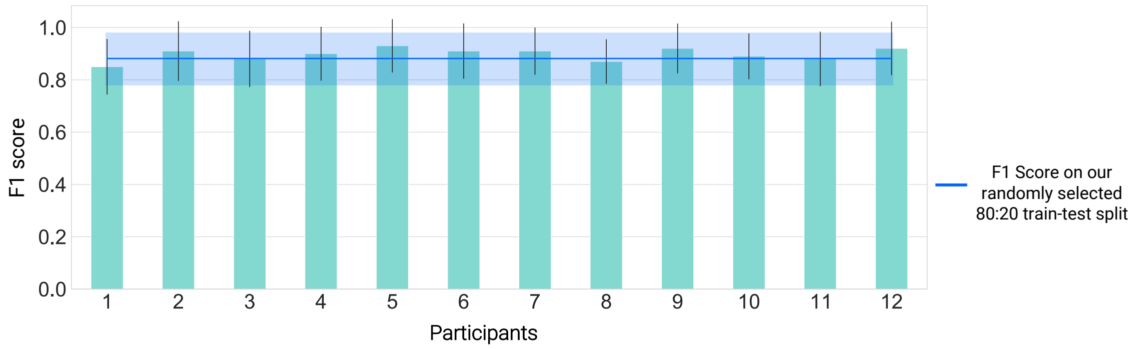


Figure 5.12: Comparison of the F1 score achieved on our randomly selected two participants with leave-one-person-out. The blue horizontal line corresponds to the average F1 score across 17 IMUs for the previous 80:20 split, and the grey band shows the standard deviation in the F1 score across all IMU counts. The vertical columns represent the average F1 score for each participant, and the error bar represents the standard deviation for each participant from count 1 to 17 IMUs.

5.2.6 Grasp-Dependent v/s Grasp-Independent Models

In our combinatorial analysis, we trained grasp-independent classifiers by combining all grasp variations. Here, we aim to investigate if these initial results can be further improved if a subset of grasps is selected. This would be relevant for application cases that comprise selected activities with a known set of grasps, or for systems that can identify the current grasp, e.g., by using activity recognition.

5.2.6.1 Method

We classified all 12 grasp variations separately (grasp-dependent models) by using our initial 80:20 split of participants' data with 19 classes [(3 fingers $\times$ 6 gestures) + 1 static hold]. To save on the computation time, we performed the full combinatorial evaluation of grasp-dependent models until IMU count = 5. There were 12 grasp variations $\times \sum_{r=1}^{5} {}^{17}C_r$ layouts = 112, 812 models.

5.2.6.2 Results

For 9 out of 12 grasp variations, the F1 score increased when the model is trained on a specific activity. Grasps like Lateral-S (Spoon), Tip-S (Needle), Lateral-L (Paper) showed an improvement in recognition of 20–30% compared to the grasp-independent model. In contrast, grasps like Cylindrical-S (Knife) and Tip-L (Pen) did not show any increment, which can be due to the object's geometry. Specifically, on such grasp variations, the fingers are tightly packed, hindering the finger movement while performing gestures.

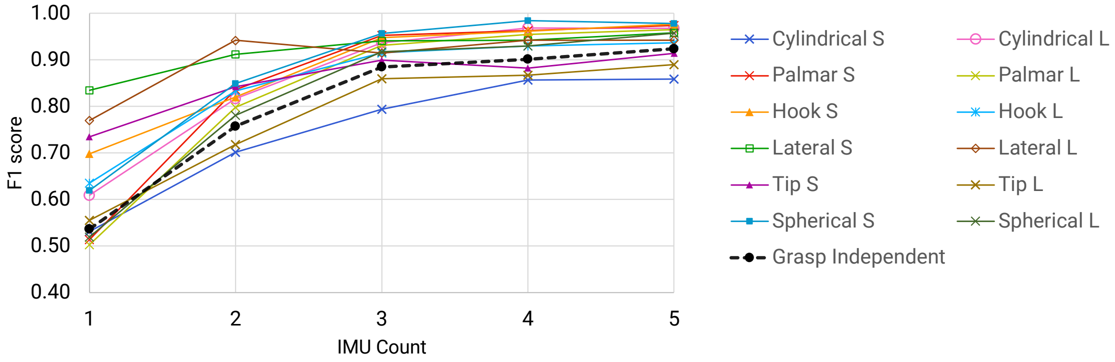


Figure 5.13: Comparison of F1 score achieved by the best layouts until an IMU count = 5 for grasp-dependent and grasp-independent models.

5.2.6.3 Implications

The performance tends to improve if the model is trained for a specific grasp variation. Therefore, when a subset of grasp-variations are chosen that map to a specific context, our results from the combinatorial analysis can further improve. This feature of selecting grasps is also integrated in our later presented design tool for finding a sparse layout.

5.2.7 Summary of Findings

The key takeaways from the above in-depth analyses are:

- *More is not always better*: Saturation in classification performance is achieved after a fixed count of IMUs as shown in Figure 5.6. In typical cases, a quite low number of 3–4 IMUs suffices for an F1 score of about 90%.

- *Possible to achieve gesture recognition via IMU on non-gesturing finger*: Our findings from placing IMUs on a non-gesturing finger in Section 5.2.4 opens up a new avenue for microgesture detection in HCI by leveraging movement patterns caused by complex hand bio-mechanics in non-instrumented fingers.

- *Effect of grasp type*: In our analysis of Grasping microgestures, we found the F1 score pattern dissimilar across different grasp variations – due to the influence of grasps on the finger pose and motions. This ultimately affects the spatial configuration of an optimal layout.

- *User-independent models*: We found that a performance of 90% and above with user-independent classification models. This demonstrates the viability of utilizing IMU-based input in future consumer-grade systems.

Given this multi-factorial design space that influences the classification performance, providing an automated system to a designer will enable rapid design iterations and decision making for optimal IMU placement. Inspired by these findings, we present a rapid technique to identify sparse layouts and a GUI-based computational design tool in the following sections.

5.3 SparseIMU: Method for Rapid Selection of Sparse IMU Layouts

Training the models for all layouts of IMUs took about 50 hours (Freehand = 1:27:31, Grasping = 22:41:52 and Freehand + Grasping = 26:20:10). Modifying the set of gestures or objects requires re-training of the models, as a new setting can influence the importance of specific IMUs. Additionally, if one wants to explore design variations, like comparing different gesture sets or sets of objects, this results in a multiplicative increase of the number of models that need to be trained and evaluated. This large computation time makes an exploratory study of IMU layouts very slow if not impossible.

To overcome this issue, we propose a method referred to as *SparseIMU*. It uses a proxy metric describing the importance of individual IMUs. As a requirement, this method should be fast to compute and correlate well with the results obtained from training all model layouts. Specifically, the proxy metric is used to derive what IMUs contribute most to the classification. In this work, we study two such proxy metrics:

- *Feature Importance*, also called Mean Decrease in Impurity [163], calculates how well a feature splits the trials into their corresponding classes. This is a natural choice for Random Forests, as the same criterion is used to build the trees themselves. Instead of training and evaluating separate models for each combination of IMUs, this approach requires training only one Random Forest model that comprises all 17 IMUs. Then Feature Importance, calculated from this model, indicates how much an individual feature is contributing. For each IMU, we use multiple features (mean, variance, etc.). Therefore, we aggregate the features belonging to the same IMU using summation to infer an individual IMU's importance. Here, the IMU with the highest importance score is essential for the classification, and the one with the lowest score contributes the least in the classification.

- *Permutation Importance* is a posthoc interpretation metric to calculate the importance of a feature. Here, a model that comprises all IMUs is trained and evaluated on the original dataset. For a specific feature, all the values in the test data are then randomly permutated; the feature, therefore, no longer provides useful information. The model is evaluated again on this corrupted dataset and the difference in performance between the original and the corrupted dataset is computed. The larger the drop in performance, the more important is the feature [164]. This approach needs no further training and only one additional evaluation for each feature. The importance of an IMU is again calculated by summing the importances of its features.

Both proxy metrices provide an importance score for each IMU. Given a desired IMU count k, one could simply choose the layout created from the top k IMUs, based on their importance score. However, in practice, it is beneficial to expand the search space of possible ”top” layouts. In particular, we search through all possible combinations of the top t IMUs (based on importance) chosen k at a time (^tC_k). We choose a t such that the total number of layouts possible with the top t IMUs (^tC_k) is at least 1% (or 10% if $k <= 3$) of the total number of possible layouts for the given count $(^{17}C_k)$ and train all those (^tC_k) models. For instance, if the desired IMU count is $k = 5$, we would choose $t = 9$, since $(^9C_5 > 0.01 \times (^{17}C_5)$ and thus we would train 126 models. Additionally, modifying this threshold of 1% allows for a user-defined trade-off between evaluation time and sparse layout performance.

5.3.1 Validation of SparseIMU Method with the Combinatorial Maximum

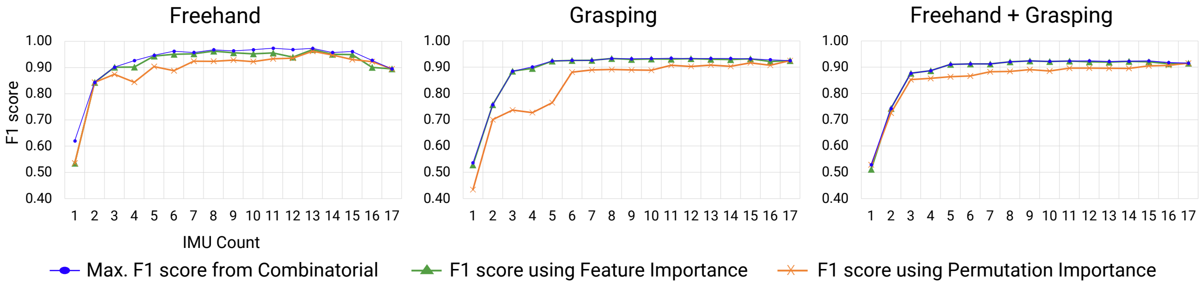


Figure 5.14: Comparison between the F1 Score of layouts from the maximum combinatorial (see Fig. 5.6) and F1 score achieved by the layouts recommended from Feature and Permutation Importance.

To benchmark the selections generated from the two proxy metrics (Feature and Permutation Importance), we use the IMU layouts from our combinatorial results that achieved the maximum F1 score in Section 5.2.2. To quantify the differences, we obtain a Spearman's correlation (ρ) between the F1 score from the max. combinatorial layout and the layouts from the two metrics. Permutation Importance received $\rho = 0.7785$ for the Freehand, 0.6617 for the Grasping, and 0.8864 for the combined condition (all p¡0.005). In contrast, Feature Importance received considerably higher correlations, with $\rho = 0.8630$, 0.9380, and 0.9419 for the respective conditions (p¡0.005). The high correlation using

Feature Importance is also visible in Figure 5.14, where the layouts consistently obtained an F1 score closer to the best performance in the combinatorial results. Therefore, we use this metric further to calculate the computation time.

5.3.1.1 Runtime

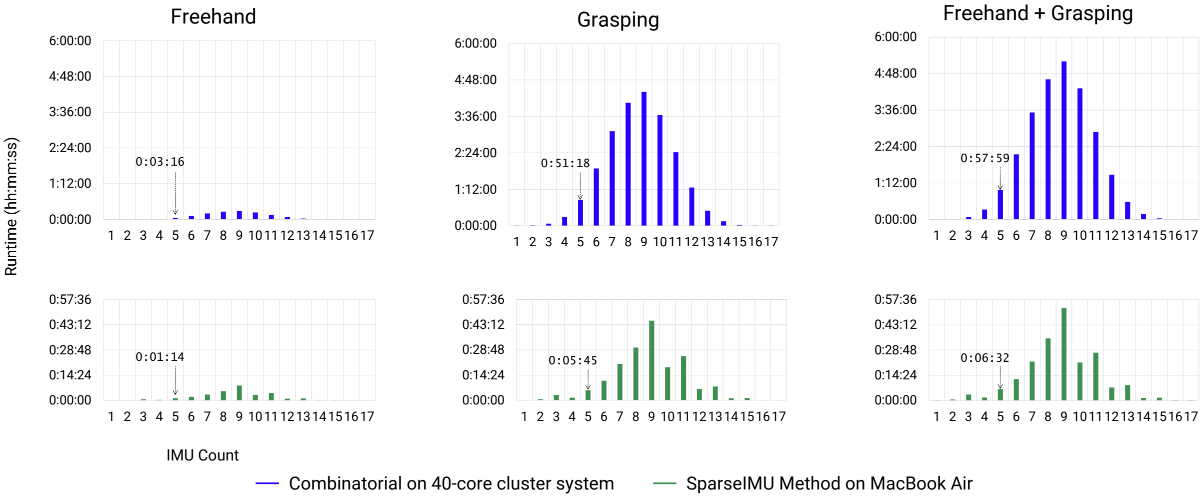


Figure 5.15: Runtime comparison between SparseIMU method and the Combinatorial Search for all three conditions: Freehand, Grasping, and Both Combined.

We now quantify the significant reduction of computation time required to select sparse layouts with the proposed SparseIMU method using Feature Importance. Given the 323K models needed to evaluate the entire combinatorial space, we used our institution's cluster system with a 40-core setup. Of note, this high-end configuration machine used in our combinatorial results is not widely accessible. In contrast, we evaluate our rapid method's performance on a commodity laptop (8-core MacBook Air). As shown in Figure 5.15, the time required to find the sparse layout by our method is significantly shorter, despite the use of a commodity laptop. This reduction is possible due to the considerably smaller number of model training required across each IMU count. For instance, if we were looking for a layout with $k = 5$ IMUs out of $n = 17$ possible IMUs in the Freehand condition, the time reduces from 3 mins on the compute cluster to 1 minute on a consumer-grade laptop. Moreover, for Grasping Microgestures and Both Combined conditions, it reduces from about 50 mins to 5 mins and from 1 hour to about 6 mins, respectively. While it takes longer to find solutions for IMU counts 7-11, we note that the method still performs significantly faster than the baseline. Moreover, we expect that layouts with this large number of IMUs need to be rarely considered, since going beyond 3–4 IMUs will only lead to a maximum increase of 4% in the F1 score, as we have shown above (see Figure 5.6). Overall, the reduction in time achieved by our method on a commodity laptop offers strong benefits for rapid iteration. In the next section, we use our method in a computational design tool.

5.4 Computational Design Tool for Rapid Selection of Custom Sparse Layouts

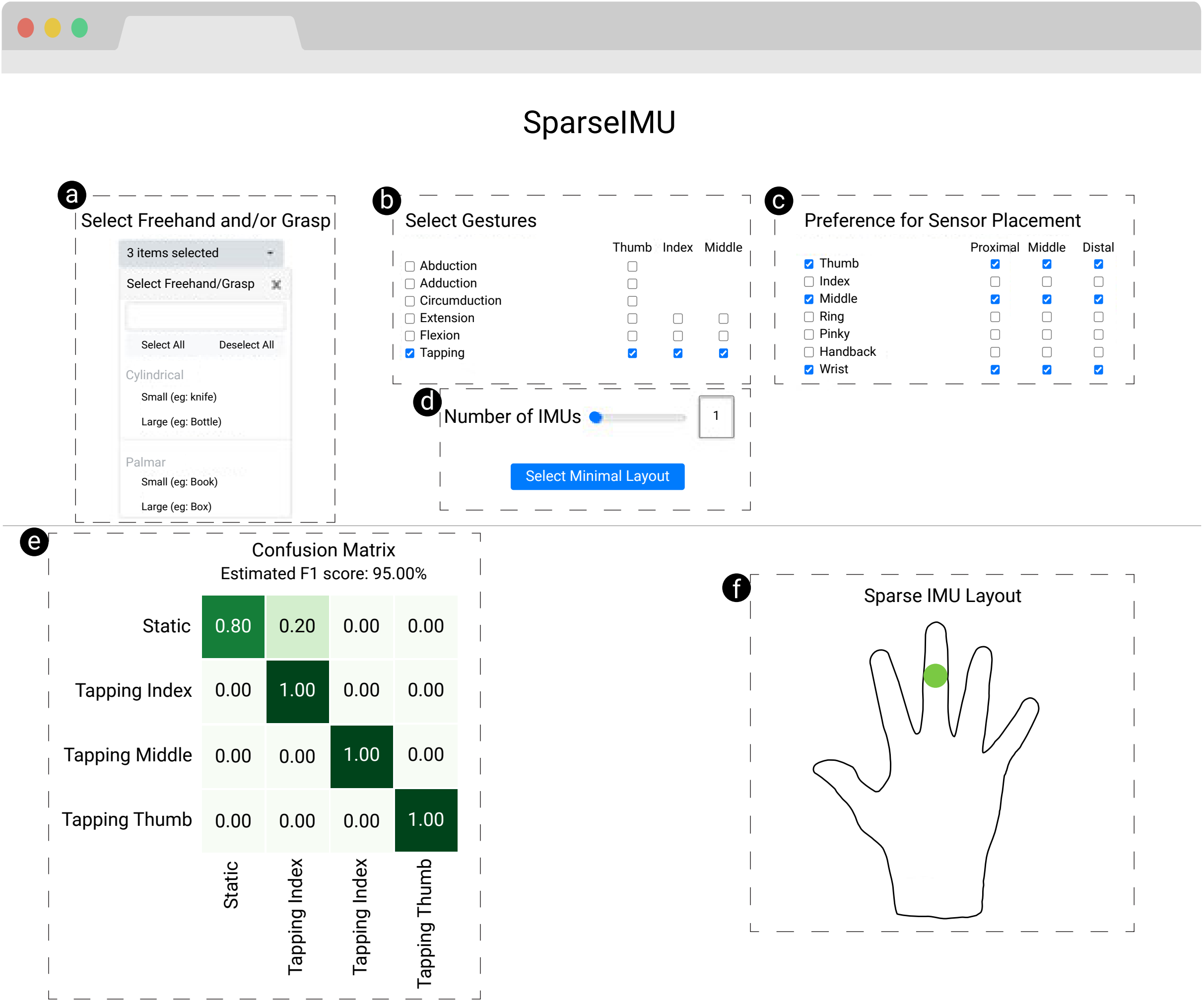
Based on the SparseIMU method for selecting IMU layouts, we contribute a computational design tool. It assists designers in the following tasks:

- *Finding a sparse IMU layout that achieves high gesture recognition accuracy:* Using the designer's specifications, the tool selects optimal designs in near real time and indicates the expected recognition accuracy. This also allows the designer to quickly obtain an initial understanding of how well a desired set of microgestures can be recognized while the user is holding certain objects. The design tool assists designers in locating fingers and precisely locating the segment of the finger where the IMU should be placed.

- *Exploring location alternatives:* Considerations of ergonomic wearability or aspects inherent to certain application cases may restrict the space where IMUs can be deployed on the user's hand. For instance, a smart ring with an in-built IMU can be more suitably placed on the ring finger than the thumb. And an application case involving dexterous manipulation of objects may benefit from IMUs placed on the proximal phalanges, rather than close to the fingertips. The tool allows the designer to restrict what locations can be augmented with IMUs, and to quickly explore alternatives.

- *Finding gestures that perform well:* While it is understood that not all gestures are compatible and will have a high performance for a specific set of objects and constraints, one key functionality of the design tool is to provide a visual representation that depicts the performance of the individual gestures. This enables the designer to quickly inspect which gestures perform well and which do not, and choose the most compatible gestures that offer high recognition accuracy.

A screenshot of the design tool is shown in Figure 5.16. The designer first selects Freehand and/or a set of Grasp variations(s) that the microgestures should be compatible with. Next, she selects the set of microgestures that shall be recognized and indicates which fingers are used for gesturing. Then, the designer can place additional constraints for IMU placement. Entire fingers or individual finger segments, as well as the back of the hand or wrist can be added or removed from the set of possible locations. As the last step, the designer selects the desired number of IMUs, to trade-off between a minimal or more



Select Freehand and/or Grasp

3 items selected

Select Freehand/Grasp

Select All Deselect All

Cylindrical

- Small (eg: knife)
- Large (eg: Bottle)

Palmar

- Small (eg: Book)
- Large (eg: Box)

Select Gestures

	Thumb	Index	Middle
□ Abduction	□		
□ Adduction	□		
□ Circumduction	□		
□ Extension	□	□	□
□ Flexion	□	□	□
☑ Tapping	☑	☑	☑

Preference for Sensor Placement

	Proximal	Middle	Distal
☑ Thumb	☑	☑	☑
□ Index	□	□	□
☑ Middle	☑	☑	☑
□ Ring	□	□	□
□ Pinky	□	□	□
□ Handback	□	□	□
☑ Wrist	☑	☑	☑

Number of IMUs 1

Select Minimal Layout

Confusion Matrix
Estimated F1 score: 95.00%

	Static	Tapping Index	Tapping Index	Tapping Thumb
Static	0.80	0.20	0.00	0.00
Tapping Index	0.00	1.00	0.00	0.00
Tapping Middle	0.00	0.00	1.00	0.00
Tapping Thumb	0.00	0.00	0.00	1.00

Sparse IMU Layout

Figure 5.16: Screenshot of the computational design tool for designing sparse IMU layouts. (a) User can select Freehand and/or multiple Grasp variations. (b) The tool automatically recommends possible gesture combinations with three fingers. (c) Additional constraints with respect to the placement of the IMUs can be specified. (d) The number of required IMUs can be selected and button click generates the results in form of (e), a confusion matrix showing the gesture-wise performance and an overall estimated F1 score, and (f), the location of the IMUs present in the sparse IMU layout.

complete instrumentation of the hand. With the click of a button, the IMU layout is then selected.

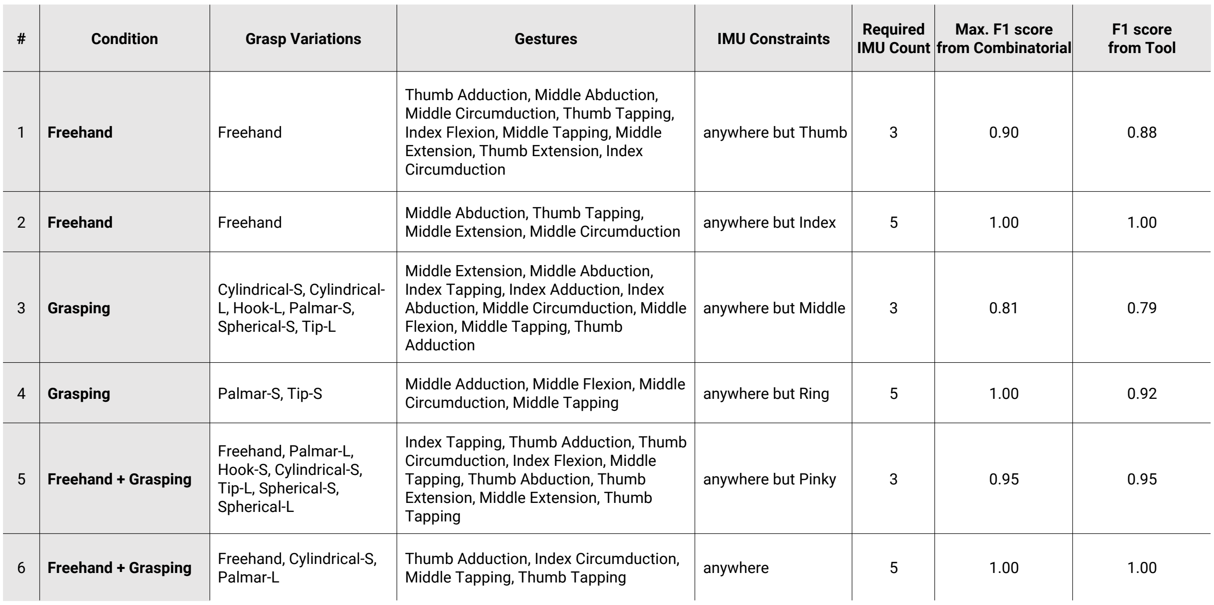
To visually present the recognition accuracy of chosen gestures, the tool displays a confusion matrix, along with location of the individual IMUs on the hand. If the designer is not satisfied with the Tool's recommendation, she can quickly explore options in an iterative manner. For instance, she may fine-tune the set of gestures or explore alternative locations for placing IMUs.

5.4.1 Implementation

It is noteworthy that our tool is different from a conventional lookup table which would require 17.5 trillions of entries to cover the various combinations of IMUs, subsets of gestures and grasp variations. Instead, by training only a few models using the SparseIMU method, our tool supports every possible custom user input while minimizing the computational complexity and storage. Furthermore, it allows the designer to rapidly iterate on multiple custom input options. Specifically, the tool uses the microgestures dataset and the SparseIMU method to identify the optimal IMU layout for a given set of requirements and constraints. The tool creates new classification models with our initial 80:20 split of train and test data. In addition to the required gestures, a Static hold is automatically added as a negative class. For generating the confusion matrix and an estimated accuracy, we use our test set. The Flask web framework for Python was used to create the tool's back-end. The front-end was styled using the Bootstrap toolkit, and JavaScript was used for client-side scripting. The Snap.svg JavaScript library was used to render the selected IMU layout.

5.4.2 Tool Evaluation

In addition to the validation of the SparseIMU method in section 5.3.1, we performed another benchmarking to compare the tool's output with the combinatorial results when the designer applies constraints and opts for choosing a subset of grasp variation and gestures. Therefore we created six example cases covering all three conditions. We randomly selected grasp variations, gestures and added finger-wise placement constraints. Informed by results from the first validation study, we chose two variations of IMU counts that we consider particularly promising for applications: 3 IMUs for a good recognition performance with very good wearability due to the low number of IMUs; and 5 IMUs for further increased recognition performance with a level of wearability that is still acceptable in many applications. We compared our tool's estimation by creating new combinatorial results for each case.



#	Condition	Grasp Variations	Gestures	IMU Constraints	Required IMU Count	Max. F1 score from Combinatorial	F1 score from Tool
1	**Freehand**	Freehand	Thumb Adduction, Middle Abduction, Middle Circumduction, Thumb Tapping, Index Flexion, Middle Tapping, Middle Extension, Thumb Extension, Index Circumduction	anywhere but Thumb	3	0.90	0.88
2	**Freehand**	Freehand	Middle Abduction, Thumb Tapping, Middle Extension, Middle Circumduction	anywhere but Index	5	1.00	1.00
3	**Grasping**	Cylindrical-S, Cylindrical-L, Hook-L, Palmar-S, Spherical-S, Tip-L	Middle Extension, Middle Abduction, Index Tapping, Index Adduction, Index Abduction, Middle Circumduction, Middle Flexion, Middle Tapping, Thumb Adduction	anywhere but Middle	3	0.81	0.79
4	**Grasping**	Palmar-S, Tip-S	Middle Adduction, Middle Flexion, Middle Circumduction, Middle Tapping	anywhere but Ring	5	1.00	0.92
5	**Freehand + Grasping**	Freehand, Palmar-L, Hook-S, Cylindrical-S, Tip-L, Spherical-S, Spherical-L	Index Tapping, Thumb Adduction, Thumb Circumduction, Index Flexion, Middle Tapping, Thumb Abduction, Thumb Extension, Middle Extension, Thumb Tapping	anywhere but Pinky	3	0.95	0.95
6	**Freehand + Grasping**	Freehand, Cylindrical-S, Palmar-L	Thumb Adduction, Index Circumduction, Middle Tapping, Thumb Tapping	anywhere	5	1.00	1.00

Table 5.1: Comparison of maximum F1 score from Combinatorial Search and Tool Output for six example cases. It includes the randomly selected grasp variations, gestures, user-defined constraints, and required IMU count. For the classification, we also had a negative class (Static hold) in each case.

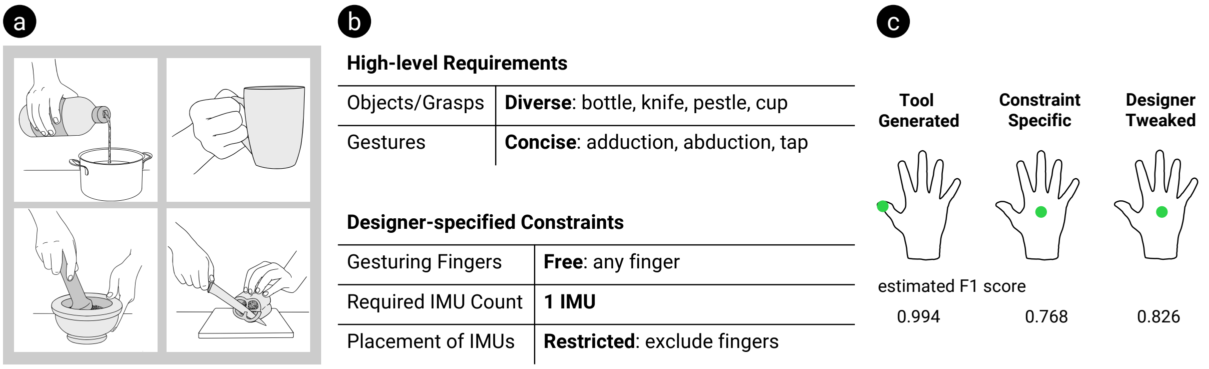
5.4.2.1 Results

Table 5.1 lists the example cases along with the results. In five out of six cases, the tool selected layouts that achieved an F1 score that was as high as the best performing combinatorial result or a maximum of 2% lower. The largest difference of 8% occurred in case 4, wherein the tool selected a layout with an F1 score of 0.92, while the best performing combinatorial layout achieved a full 1.00. Noteworthy, the tool also performed well in case 3, in which most of the randomly selected gestures involve the Middle finger whereas the constraint was to exclude the Middle finger from placing IMUs. Despite this demanding constraint, the tool successfully selected a layout that achieves performance close to the layout found by exploring the entire combinatorial space.

5.5 Application Scenarios

In this section, we present a set of four scenarios, each illustrating a realistic application of freehand and grasping microgestures with different design requirements and constraints. We demonstrate how our computational design tool can assist designers in deciding between various layouts, which is a non-trivial problem potentially requiring a trade-off, and can help in refining IMU-based sensing solutions.

5.5.1 Kitchen: Supporting Diverse Objects with Minimal Instrumentation



High-level Requirements

Objects/Grasps	**Diverse**: bottle, knife, pestle, cup
Gestures	**Concise**: adduction, abduction, tap

Designer-specified Constraints

Gesturing Fingers	**Free**: any finger
Required IMU Count	**1 IMU**
Placement of IMUs	**Restricted**: exclude fingers

Figure 5.17: Supporting diverse objects a) with minimal instrumentation b) in a smart kitchen scenario requires a trade-off between F1 score and IMU postion c).

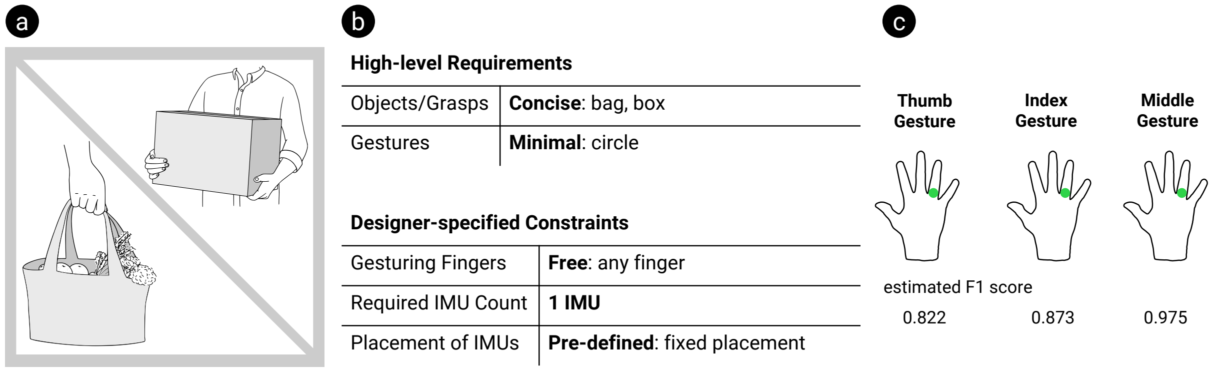
Smart kitchens, providing in-situ instructions while cooking, have been a popular research area over the last decade [165]. We envision our computational design tool to support a designer, Alice, in the development of an in-situ recipe manager that supports information access using microgestures while cooking. For her first prototype, Alice wants to enable microgestures on *four objects* commonly found in the kitchen: knife, bottle, cup and pestle (cf., Fig. 5.17-a). For browsing a recipe, her application requires a *small, concise set of gestures*: back (abduction), forward (adduction) and select (tap). Due to frequent hand washing, the layout should be minimal (1 IMU) and restricted to the back of the hand or wrist (cf., Fig. 5.17-b).

Tool Output: With the selection of objects and gestures (and no further constraints imposed), the computational design tool suggests the thumb as common finger capable of performing all desired gestures, and the thumb's middle segment for IMU placement. Being 'most ideal', this sensor location achieves an F1 score of 99.4% (cf., Figure 5.17-c). However, Alice, excluded the fingers as sensor locations for sanitary reasons. This restrains sensor placement to the back of the hand and wrist, which achieve an F1 score of 76.8% and 56.6% respectively. For both, the confusion matrices reveal that the adduction gesture has a lower score, likely due to the large distance between the IMU and the gesturing finger. As a result, Alice settles on a trade-off between IMU location and available gestures. To keep the IMU position on the back-of-the-hand, she updates her design to include only tap and abduction gestures, increasing F1 score to 82.6%.

5.5.2 On-the-Go Interaction

5.5.2.1 Sensor Placement on Non-Gesturing Finger

As voice user interfaces are oftentimes prone to false activation [166], wake-gestures are an attractive remedy [114, 167, 168]. Bob aims to explore wake-gestures that work in on-the-go scenarios where both hands are occupied, e.g., while carrying two bags or a box (cf., Fig. 5.18-a). Furthermore, he intends to leverage an existing smart ring that he intends to 'hack' to access its IMU data. It does not matter which finger performs the gesture. However, ideally, the ring would keep its current position: worn on the ring finger's proximal segment.



High-level Requirements

Objects/Grasps	**Concise**: bag, box
Gestures	**Minimal**: circle

Designer-specified Constraints

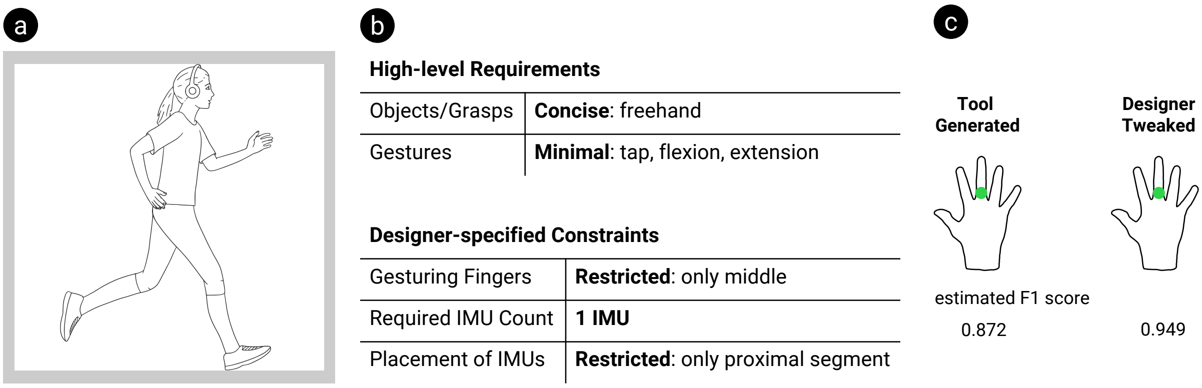
Gesturing Fingers	**Free**: any finger
Required IMU Count	**1 IMU**
Placement of IMUs	**Pre-defined**: fixed placement

Figure 5.18: An on-the-go scenario a) with pre-defined sensor placement on a non-gesturing finger b) leverages co-activation c).

Tool Output: Bob starts by evaluating the circle gesture performed with the thumb and the IMU present on the ring finger. The tool outputs an F1 score estimate of 82.2%. As wake-gestures should be resilient to false activation, Bob is not satisfied yet and explores further possibilities. As the position of the IMU is non-negotiable, he includes index and middle as gesturing fingers which achieve an F1 score of 87.3% and 97.5% respectively. The middle finger's promising performance (97.5%) is explained with the higher co-activation sensed on the ring finger (where the sensor is worn). Here, the computational design Tool allowed Bob to iteratively explore the gesture space and finally arrive at a tailored solution.

5.5.2.2 Finding Unambiguous Combination of Gestures

Listening to music while running is a typical combination, but controlling the music app on a smartphone or smartwatch's touchscreen requires Taylor, a frequent runner, to take unplanned breaks as shown in Figure 5.19-a. Conventionally, she needs to pause her run for performing the desired command (switch tracks or play/pause). These frequent and unnecessary halts for simple inputs affect her lap timings. She would prefer to use her middle finger for gesturing since she keeps switching the index and thumb poses in different fist forms while running. Her requirements are only for three gestures, including Tap, Flexion, and Extension. Also, due to vigorous hand movements and to keep the IMU firmly attached to her finger, she chooses to place the IMU ring in the proximal segment, which can be on any finger (see Figure 5.19-b).



High-level Requirements

Objects/Grasps	**Concise**: freehand
Gestures	**Minimal**: tap, flexion, extension

Designer-specified Constraints

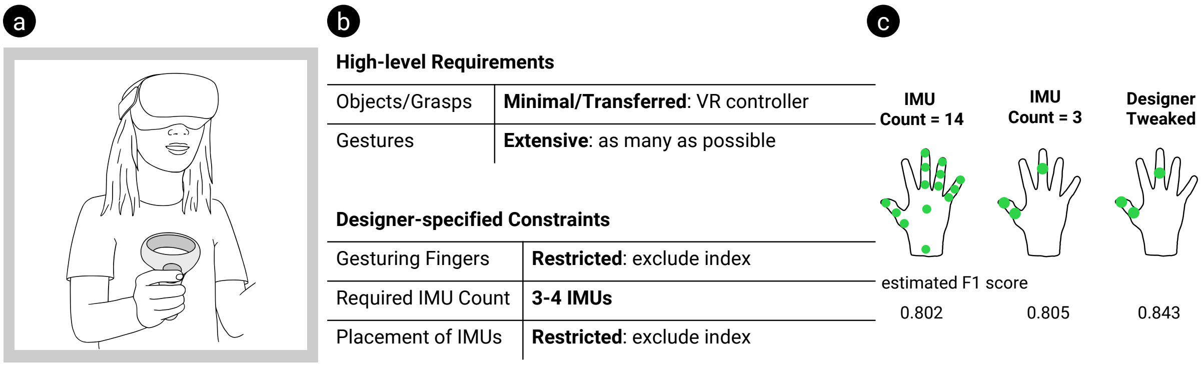
Gesturing Fingers	**Restricted**: only middle
Required IMU Count	**1 IMU**
Placement of IMUs	**Restricted**: only proximal segment

Figure 5.19: Supporting Freehand a) with minimal but clearly distinguishable gesture set b) in a running scenario with a restricted placement choice c).

Tool Output: Taylor started by opting for Freehand gesture and then made her gesture choices, and selected all fingers' proximal segment. As one's intuition, the tool suggested placing the IMU on the Middle Finger's proximal segment. It predicts an estimated score of 87.2%. By analyzing the confusion matrix, Taylor found out Flexion and Tap gestures get confused and subsequently decided to find the performance of other gestures. Using the rapid evaluation provided by the tool, she found out that replacing Flexion with Abduction solves this issue, and an estimated F1 score of 95% is possible (see Figure 5.19-c). Here, the tool was beneficial in finding an alternative gesture that can be detected at a higher performance while preserving all the other requirements.

5.5.3 VR Controller: Diverse Gestures with Minimal IMUs

Exploring diverse gestural inputs for VR [169] has been a popular area for experimentation in HCI and media arts. Dan plans a VR media arts installation which uses microgestures on a hand-held VR controller to contrast private and public interactions by subtly expanding the controller's range of functions. Thus, as demonstrated by [170], he aims for a miniaturized device equipped with 3–4 IMUs in combination of a commodity VR controller. He wants to avoid placing IMUs on the index finger which operates the VR contoller's push button and also not use it as a gesturing finger. To facilitate playful public or private interactions, he hopes to support as many different gestures as possible.



High-level Requirements

Objects/Grasps	**Minimal/Transferred**: VR controller
Gestures	**Extensive**: as many as possible

Designer-specified Constraints

Gesturing Fingers	**Restricted**: exclude index
Required IMU Count	**3-4 IMUs**
Placement of IMUs	**Restricted**: exclude index

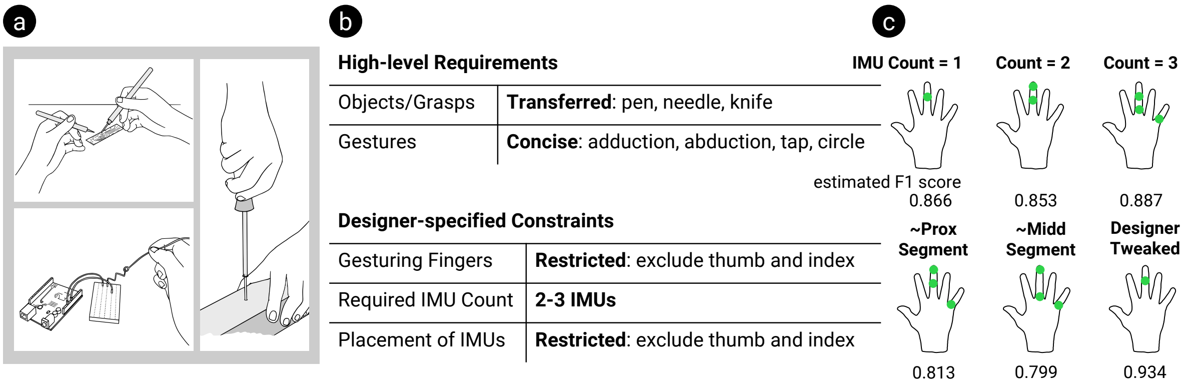
	IMU Count = 14	IMU Count = 3	Designer Tweaked
estimated F1 score	0.802	0.805	0.843

Figure 5.20: Minimal setup with 3–4 IMUs a) with maximum diverse set of gestures b) finding the balance between gestures and accuracy.

Tool Output: Dan explores the solution space for all possible IMU locations excluding the index finger (14 IMUs total). The tool yields an F1 score of 80.2% if 12 gestures are supported. Dan iteratively decreases the IMU count (while keeping the amount of gestures to 12) inspecting performance after each decrement. He identifies a saturation in F1 score at 3 IMUs (80.5%), which illustrates that a higher number of IMUs does not necessarily imply better performance (cf., Fig. 5.20-c). After further tweaking their configuration, Dan settles on a 3-IMU configuration and a set of 10 gestures. This choice is a trade-off allowing for a relatively high amount of gestures while still achieving an F1 score of 84.3%. As Dan aims for a rather playful, explorative VR installation, he considers this level of score acceptable. This highlights how the choice of a final layout depends on the weight the designer assigns to the different parameters (e.g., amount of gestures vs. performance) which in turn strongly relate to the specific application (e.g., playful vs. safety-critical purposes).

5.5.4 Electronics Workshop: Microgestures while Performing High-Precision Tasks

Carla seeks to explore how users can make use of microgestures to access additional instructions during high-precision tasks such as soldering. She envisions tools such as a soldering iron, soldering lead, or a screwdriver (cf., Fig. 5.21-a). As these tools are not available in our dataset, she uses our computational design tool to make an informed best guess by determining a set of initial layouts to elaborate on. Here, our Tool draws strength from the similarity in grasp types: the soldering iron (not present in the dataset) is typically held in a fashion similar to the pen (present in the dataset); holding fine soldering lead or wire in place resembles holding a needle, and holding a screwdriver demonstrates a similar (cylindrical) grasp like holding a knife. Carla envisions four gestures: forward, backward, select, and circle which she intends to use to browse an instruction manual. She furthermore excludes thumb and index finger–both as gesturing fingers and for IMU placement–to not interfere with the high-precision soldering task, and constrains the number of IMUs to 2 or 3 (cf., Fig. 5.21-b).



High-level Requirements

Objects/Grasps	**Transferred**: pen, needle, knife
Gestures	**Concise**: adduction, abduction, tap, circle

Designer-specified Constraints

Gesturing Fingers	**Restricted**: exclude thumb and index
Required IMU Count	**2-3 IMUs**
Placement of IMUs	**Restricted**: exclude thumb and index

Figure 5.21: Transfer of grasps a) with restrictions on Thumb and Index b) finding the optimal finger segment c).

Tool Output: The computational design tool suggests placing the IMUs on the middle finger which achieves a competitive F1 score of 88.7% when 3 IMUs are used. Yet, at closer inspection, the tool also reveals that accuracy varies depending on the finger segment on which the IMU is placed, ranging from 80% to 88%. Hence, the choice of finger segment is crucial. Moreover, the tool shows that there is only 2% gain in score from placing 3 IMUs on the middle and pinky finger (88.7%), compared to only one IMU on its middle segment. Thus, a single IMU is sufficient to cover all gestures Carla had planned for her scenario. Further exploration shows that an increase in accuracy can be obtained for the 1-IMU layout to 93.4% by removing the adduction gesture (cf. Fig. 5.21-c). As follow up, Carla conducts a small-scale data collection using the 1-IMU layout recommended by the tool. Here, the tool provided a best guess in terms of IMU placement and gesture choice which served as a strong foundation for further iterations.

5.6 Comparing the Tool’s Output with Live Gesture Recognition

To further demonstrate the tool’s practical usefulness and generalizability to real-world applications, we collected another dataset with different hardware configurations and participants. This section compares the predicated F1 score from the computational tool with another system deployed for live gesture recognition.

5.6.1 Apparatus

With a focus on mobility and wearability, we developed a working wireless system that consists of a 9-Axis IMU (MPU9250, InvenSense Inc., CA, USA) and a Bluetooth module.

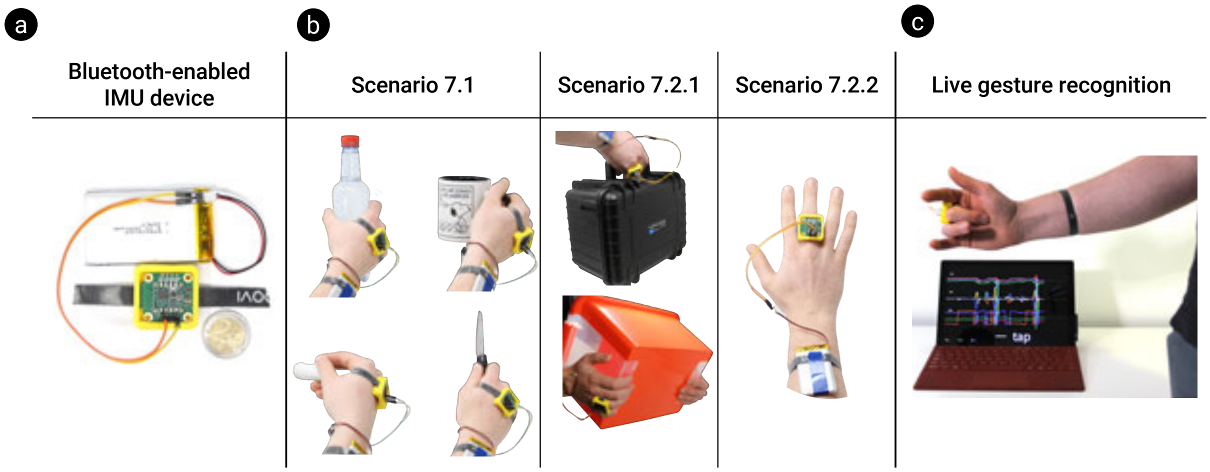


Figure 5.22: Minimal wireless hardware with battery a); scenarios involving multiple objects and freehand b); live classification of gestures c).

As with previous work for gesture detection with a low-power wearable device [171], we sampled the accelerometer at 35 Hz (lower than in our microgestures dataset). Similarly, gyroscope and magnetometer were sampled at 35 Hz. For powering the device, we used a 2000mAh (DTP634169) lithium polymer battery. We also created a 3D printed casing with hooks to attach velcro straps so that the device can be easily worn on different fingers and varied hand sizes. An additional velcro strap and adhesive tape were used to affix the battery to the arm such that it would not interfere with hand actions. We created two such devices (as shown in Figure 5.22-a) and synchronized them to enable data collection from multiple hand segments simultaneously. Raw data from the devices is wirelessly streamed over Bluetooth to a PC for live classification.

5.6.2 Scenarios

To keep the data collection feasible, we selected three scenarios from Section 5.5.1, 5.5.2.1 and 5.5.2.2. These represent multiple settings with gestures on diverse objects, on-the-go interaction with sensor placement on the non-gesturing finger, and finding an unambiguous combination of gestures for freehand input, as shown in Figure 5.22-b.

5.6.3 Participants

We recruited 6 right-handed participants (3M, 3F, mean age: 22.2; SD: 2.5) with an average hand sizes from Wrist to the tip of Thumb = 132mm (SD:9mm), Index = 168mm (SD:10mm), Middle = 175mm (SD:12mm), Ring = 163mm (SD:10mm), Pinky = 144mm (SD:10mm). It is noteworthy that all 6 participants were different from those who participated in creating the microgestures dataset (section 5.1.4).

5.6.4 Task and Procedure

We used the same procedure as described in Section 5.1.5 i.e. we counterbalanced the two conditions (Freehand and Grasping) and further counterbalanced the order of objects in each scenario. Once the object or freehand condition was selected, we presented the gesture/non-gesture states in a randomized order. We developed a custom software tool using Flask framework in Python to label the trials that the experimenter controlled during data collection. Overall, we recorded 5 trials for each gesture and Static hold for a negative class, totaling 870 trials (145 trials per participant), comprising 10 unique gestures and static hold classes on 7 different object/grasp types.

To evaluate a potential bias resulting from orientation, the data collection for this experiment was performed in a room that was different from the microgestures dataset. Additionally, the orientation of the participants was rotated by 90 degrees left from their original orientation in the microgestures dataset. The sitting/standing posture and the start and stop for labeling were similar for all scenarios as in the microgestures dataset, except for the scenario with freehand gestures (Figure 5.22-c). Here, we kept the posture to standing as defined in the scenario and marked the start and stop of gestures when the arm started swaying upwards from the standstill posture and returned to the initial state. Hence, the assumption is that even though coarse hand movement is involved, IMU placement is still crucial for detecting fine finger movements (gestures). The complete data collection for each participant took about 45 minutes.

5.6.5 Feature Extraction and Classification Model

In order to perform a systematic comparison, we extracted the same six features as used in the analyses above and in the computational design tool. These features are mean, median, minimum, standard deviation, and variance calculated from each of the 9-axis of the IMU. It is important to note that live classification requires a time window of streamed data as opposed to our tool in which we classified the entire trial. Therefore, the features were extracted on a window size of 90 and an overlap of 70 frames - only for the data collected in this study. The tool configurations remain untouched, which extracts features over the entire trial. We also used the same classifier with default parameters as used in our computational tool, i.e., Random Forest (RF) with max_depth = 30. We trained a separate grasp-independent multiclass model (not encoding grasp/object information in the class labels but only gestures) for each scenario and IMU placement. Since our participant count is lower than in the microgestures dataset, in addition to the *user-independent* models with leave-one-person-out cross validation training and testing, we also created *user-dependent* models and evaluated with leave-one-trial-out cross validation technique.

5.6.6 Results

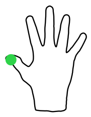
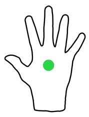
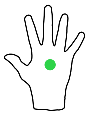
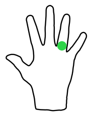
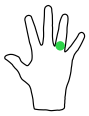
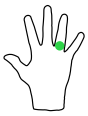
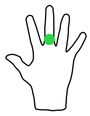
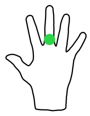
		Scenario 7.1 Supporting Diverse Objects with Minimal Instrumentation			Scenario 7.2.1 Sensor Placement on Non-Gesturing Finger			Scenario 7.2.2 Finding Unambiguous Combination of Gestures	
Objects/Grasps		bottle, knife, pestle, cup			bag, box			freehand	
Gestures		Thumb Adduction, Thumb Abduction, Thumb Tap	Thumb Adduction, Thumb Abduction, Thumb Tap	Thumb Abduction, Thumb Tap	Thumb Circle	Index Circle	Middle Circle	Middle Tap, Middle Flexion, Middle Extension	Middle Tap, Middle Abduction, Middle Extension
IMU placement									
Computational Design Tool		100% I 0.994	77% III 0.768	83% II 0.826	84% III 0.822	90% II 0.873	100% I 0.975	92% II 0.872	100% I 0.949
Live Classification	User-independent	100% I 0.860	67% III 0.577	79% II 0.682	93% III 0.903	98% II 0.953	100% I 0.973	95% II 0.615	100% I 0.644
	User-dependent	100% I 0.902	95% III 0.857	98% II 0.886	98% III 0.949	99% II 0.959	100% I 0.969	96% II 0.820	100% I 0.857

Table 5.2: Comparison of F1 scores from the computational design tool output and live classification. For each of the three scenarios, the object/grasp information, gesture, and location of IMU placement are described. We also included a negative class (Static hold) wrt. objects/grasps. For each scenario, the normalized F1 score of a configuration is calculated by normalizing it to the highest achieved F1 score. For completeness, we also report the absolute F1 score obtained for each configuration below the ranking. The performance ranking is denoted in roman characters.

Table 5.2 shows the comparison between the estimated F1 score from the computational tool and the performance achieved in the live classification. To understand the relative performance across configurations within a scenario, we calculate their normalized F1 score. The normalized F1 score is calculated by normalizing the F1 score of a given configuration with respect to the highest-performing configuration within this scenario. The table shows the normalized F1 scores (represented as percentages) along with absolute values for completeness. We observed that even with different hardware and participants, the results for live recognition are in congruence with the tool's prediction. Specifically, the tool correctly predicts the performance ranking of configurations, and the normalized F1 scores across configurations matches reasonably closely. Of course, this does not hold true for the absolute values, which strongly depend on the (largely differing) settings of a configuration (live classifier, different hardware, model, train trials). However, the normalized F1 score gives an indication of what changes (improvement or deterioration) to expect when switching from one configuration to a different one. It is noteworthy that our results are consistent for all three scenarios with user-dependent as well as independent models, demonstrating the generalizability of our method.

5.7 Discussion and Limitations

While this work takes a significant first step toward the rapid dense-to-sparse exploration of IMU layouts for finger microgestures, we mention our work's strengths and limitations below:

5.7.1 Grasps, Objects, and Gestures in and beyond the Microgestures dataset

When constructing our dataset, we leveraged prior work on grasp types [15] to build six categories and selected representative small and large objects as well as corresponding realistic actions (cf., Figure 5.3). While exhaustively covering all conceivable objects for each grasp type is impossible, we anticipate generalizability for objects not present in the dataset. A few characteristics of finger movements directly depend on the grasp type and hence generalize for objects beyond the ones present in the dataset, such as the feasibility of gestures with a specific finger, and the co-activation of the non-gesturing finger. There are few other characteristics of object manipulation which might not generalize and which future work needs to address. For example, two objects may afford the same grasp type but fulfill different purposes (e.g., pen vs. soldering iron) and require different movements (fluent writing vs. a steady hold for soldering). Our computational design tool incorporates this limitation by assuming the user would briefly pause the primary activity while keeping the object in hand.

5.7.2 Computing, Refining, and Transferring Layout Suggestions

We anticipate that the tool's layout suggestions can serve as a valuable starting point to quickly reduce the design space and for further improvement of performance in an end-to-end working system. Additional techniques such as collecting more training data to include additional variations, adding more features, performing hyperparameter tuning to tailor the classifier's behavior to the specific dataset, creating an ensemble of classifiers, and optimizing the hardware's sample rate to improve the recognition rate can be applied, if desired. Our findings show that grasp-dependent models may further improve the classification performance. This also suggests that the combination of target Freehand and/or Grasp variations affects the model's performance, where our computational design tool can be useful in rapid testing and iterations to find the balance between users' choice and classification performance.

While we performed user-independent evaluations in our analysis, in our initial tests, we found the performance of user-dependent models is higher with the same model architecture. With the advances in deep learning models and their interpretability methods, we believe a more sophisticated model pipeline can be constructed based on our analysis results. This would also help researchers in benchmarking different techniques to select sparse layouts.

5.8 Conclusion

In this work, we presented the first computational design approach for realizing sparse IMU layouts to recognize microgestures effectively – with hands-free and while holding everyday object conditions. Our *SparseIMU* method uses a customized version of a well-known ML metric (Feature Importance) for rapidly selecting sparse IMU layouts. We also contributed a computational design tool that selects sparse IMU layouts based on higher-level inputs (objects, gestures) and constraints (e.g., choice of placement) specified by the designer. We empirically validated the accuracy of the IMU layouts selected by our design tool with the combinatorial results obtained by training 393, 213 models. Selecting a sparse layout with our SparseIMU method is significantly faster than exploring the complete combinatorial space and shows a high quantitative agreement. We also contribute the first microgestures dataset, consisting of 18 gestures and 3 non-gesture states performed with freehand and 12 objects covering all the six grasp types. Using a dense network of 17 synchronized IMUs placed all over the dominant hand, we collected the data from 12 participants. Our dataset comprises fully annotated dense IMU data consisting of 13,860 trials (3 million frames). Through our dataset, we believe new insights can be derived not only for HCI research but might also be helpful for an array of other fields, including machine learning, optimization and bio-mechanics.

Our analysis revealed three major findings: i) With only 3–4 IMUs, an F1 score of about 90% can be achieved in a challenging classification task with 18 classes of Freehand and Grasping microgestures, ii) placing an IMU on a different segment on the same finger may significantly affect the classification performance, and iii) we demonstrated the feasibility of detecting gestures with an IMU placed on a non-gesturing finger. Finally, through a set of systematically designed application cases and a user study, we demonstrate how our computational design tool enables designers to employ a rapid and iterative design process for realizing microgestures for diverse scenarios across multiple objects. Our contributions in this thesis take advantage of fingers' dexterity and uncover the sensing potential of IMUs towards bringing computing at user's fingertips – practically everywhere and always.

CHAPTER 6

Conclusions and Future Work

Human beings now frequently need to interact with computers in their professional tasks as well as leisure activities. This need has steadily grown with the proliferation of mobile devices. However, current computing devices include touch screens, buttons, or controllers, which require at least one hand free for input. As a result, interaction with the digital world is rendered challenging when the hands are occupied with other everyday tasks. Therefore, alternative input modalities are required to support interaction during other hands-busy scenarios. These practical input techniques must minimise recognition errors, but they must not be so restrictive as to hinder productivity. Instead of adding a "touch screen" to the held objects, this thesis attempts to meet this need by investigating ways to design and implement seamless gestures for always-available input. In particular, I address the three fundamental research questions as outlined in the introduction:

1. How does the multitude of grasp types, and object geometries affect users' choice of microgestures?
2. How to avoid false activations in gestural input while handling everyday objects?
3. What sensor locations on the hand provide effective recognition with minimal instrumentation?

This last chapter summarizes the contributions to the above research questions and concludes by providing directions for future research.

6.0.1 Insights into how end-users perform gestures when holding an object

In Chapter 3, I conducted the first study to answer the essential question of how grasps and object geometries affect the design space of microgestures performed on handheld objects under the interactional constraints caused by holding a physical object in one's hand. In particular, using a taxonomy of six different grasps and two object sizes, I selected 12 representative handheld objects from various domains. The study employed the user elicitation method to analyze over 2,400 user-generated microgestures for ten referents on all objects, which allowed for identifying user agreement, mental models, and gesture preferences. From this data, I characterize users' preferred type of gestures when hands are busy. I also show that these gestures mainly depend on the referent rather than the grasp or object, but that the choice of fingers and gesture location is strongly influenced by the size and grasp type of the object. Finally, using statistical clustering, I derive a new class of gestures called *Grasping Microgestures*, which prescribes a starting point for consolidated gesture set that is compatible with diverse objects geometries.

6.0.2 Simple, robust and scalable interaction technique

Many existing gesture design approaches are only concerned with ergonomics or technical feasibility. The issue of false activation, though severe, has been a long-standing and most largely unaddressed concern even in massively deployed systems [172]. To address the problem of false activation during input while grasping everyday objects, I designed and validated a novel concept called *SoloFinger* in Chapter 4. The main finding of this work is that single finger movements that are rapid, easy, and elegant to perform can indeed function as robust microgestures while holding objects. I demonstrate that this holds true for diverse grasps, object geometries, and everyday actions. Specifically, SoloFinger leverages the insight that fingers tend either to be static or to have multiple fingers move concurrently when holding and manipulating objects. Consequently, a single finger's extensive yet comfortable movement while other fingers remain idle stands out from everyday hand-object motions. Through this technique, designers can create robust gestures with everyday objects without affecting their intrinsic properties. To methodologically validate this idea in the context of hand-object manipulations and gesture design, I conducted an extensive data collection study. I performed a series of data-driven analyses, including the concept's validation with a pre-existing dataset. Additionally, I implemented a proof-of-concept system a commercially available VR glove and a multi-class classification of seven SoloFinger gestures with the thumb, index, or middle finger, achieving an accuracy of 86% with a very low number of false activations (17 out of 800 trials). When the held object is known, the accuracy increases to 89%, with no false activations in the collected dataset.

6.0.3 Rapid computational method for selecting sparse sensor layouts

Inertial Measurement Unit (IMU) sensors have shown promising results for gesture recognition in an ergonomic and lightweight ring form factor. Aside from their sensitivity to subtle movements, they do not face occlusion problems. However, the IMU layout, i.e., where IMUs are placed on the hand and fingers, is very important for detecting gestures accurately [89]. IMU placement is subject to a multi-factorial design space that includes freehand or grasping conditions, diverse object geometries, different fingers, various gestures, and additional user-defined constraints. Finding an optimal layout manually while considering these factors is extremely time-consuming and might result in layouts that are far less than optimal. The *SparseIMU* method, presented in Chapter 5, takes the first step toward using computational methods to rapidly select minimal IMU layouts for gestural input with IMU sensing. In essence, SparseIMU utilizes a modified version of feature importance to select layouts rapidly. With a dense network of 17 IMUs placed on the hand, I collected data from multiple microgestures in freehand and grasping conditions. Our empirical analyses included evaluating the entire combinatorial space of 393K IMU layouts and comparing various IMU layouts. I also developed a GUI-based tool utilizing the SparseIMU method for designers of gesture recognition systems to make well-informed and rapid decisions. Last but not least, another user evaluation with a separate IMU hardware confirms that the tool's predictions and performance achieved in live gesture recognition are congruent.

6.1 Future Work

In keeping with the original Mark Weiser's vision of ubiquitous computing [173], I envision a future where the user takes advantage of fingers' dexterity to enable computing at the right time and place. While this thesis focused on addressing three major challenges to enable always-available input, the following interdisciplinary endeavors can further unlock new user experiences in varied domains like healthcare, automation, construction, smart homes, and many more.

6.1.1 Expressive, domain-specific gestures and rethinking object designs

The extensively studied gestures in this thesis mainly consist of taps and swipes. With the idea of adding expressivity, future work can create new expressive gestures like squeezing or moving the object in a particular manner that resonate well with the object in hand.

Moreover, the gesture designs presented in this thesis with Grasping and SoloFinger microgestures, require holding an object in a static position. It is important to note, however, that the analysis of false activation *did* involve object manipulation in Chapter 4.

Future work should explore the effect of simultaneously performing a gesture while manipulating an object, such as performing an input gesture while hammering. In this dissertation, I primarily focused on the grasp types and then included multiple variations of everyday objects (over 50 entities) for analysis. With grasp types as a vantage point, our findings are more generalizable and could be applied to new objects, presenting several unexplored opportunities to enable gestures while holding objects. While there can be so many possibilities, future work can take inspiration from the factors described in Wimmer [174] GRASP model and evaluate these gestures in specific scenarios. For instance, adding interactivity to the assistive equipment (e.g., walking cane), building new input techniques where the Artificial Intelligence (AI)-enabled modalities (such as voice assistants) can cooperate to make the best use of the object in hand, and increasing the number of default functions on controllers in immersive environments like virtual or augmented reality. Most often, the design of everyday objects remains the same over time, and users have no other option but to accept the design flaws. Future designers could augment/alter existing objects to integrate gestures ergonomically and combine other object properties to reduce the overhead of carrying multiple things simultaneously.

6.1.2 Capturing diverse and long-term datasets

The two large datasets released, especially in combination with the activity data in Chapter 5, can be used in future work for Transfer Learning [175] as both gesture and non-gesture conditions are present. Moreover, future work may choose to augment the released datasets with additional objects and activities or gestures. For example, there is potential to expand into rhythmic gestures incorporating longer duration or repetitive gestures (e.g., double taps), which indicate benefits such as robust wake gestures or hot words [176]. It will also be relevant to study objects with advanced material properties, such as pronounced surface texture, friction, or deformability. For feasibility reasons, the Microgestures dataset contains gestures performed by Thumb, Index, and Middle fingers, but future work should investigate gestures performed by other fingers. In addition, the dataset was collected from participants who were right-handed and young. Future work may study how this data generalizes to other populations such as the elderly (potentially limited range of motion or tremor) or children (smaller hands). For creating the datasets, I have carefully selected different object geometries that afford different orientations of hand and fingers to reduce potential dataset bias. For instance, the thumb faces upwards while holding the book, but it is sideways while holding the bottle. As a next step, future work may use data augmentation techniques to arbitrary facing (or even orientation) of the head by adding randomized orientation offsets to the raw data [177].

In Chapter 4, I tested false positives on an extensive dataset covering a large set of grasps, objects, and actions. However, the dataset is subject to limitations. The reason for using this dataset rather than a field recording was that it offered precise and realistic hand data with a broad range of hand-object actions. Given the technical limitations of recording such highly articulate actions, this would likely be challenging in a field recording. Specifically, the dataset contains focused activities, which I worked on as a first step to exploring our concept's potential. However, in addition to the focused activities, different idle phases may occur in a real-world interaction. Idle phases might bring additional challenges for classification, for instance, during change of hands, multi-tasking, nervous tapping, or when the user is fiddling with an object. These should be investigated in future work by capturing longer-term in-the-wild data.

6.1.3 Interpretable, low-resource recognition models

Most state-of-the-art gesture recognition systems are opaque and do not provide reasoning when the classification succeeds or fails. The recognition systems should be designed in a way to assist engineers in understanding the recognition failures and fine-tune models for failure cases. This knowledge will help in the mainstream adoption of gesture recognition systems. In this dissertation, I have started to address this problem by creating interpretable systems in SoloFinger 4, wherein I used model interpretability to recognize the reasons for false activations. Since I used one set of thresholds for all users, it will need to be investigated whether this generalizes to children or users with very large hands. Future work could normalize thresholds for hand size or finger length and use them in deployment.

In addition, learning-based recognition approaches require a lot of training data that is difficult to collect. Data collection and labeling is a well-known problem in HCI and Machine Learning, but the manually-labeled frames in the released datasets of this thesis can provide a quality source for auto-labeling of new data, reducing the tedious manual efforts of data labeling. Finally, it is worthwhile mentioning that the datasets offer a starting point to enable always-available input using IMUs, but it would be fruitful if future works investigate effortless methods for data collection and labeling in the wild.

6.1.4 Computational tools to assist designers and engineers

Currently, the computational tool in Chapter 5 suggests sensor placement based on gestures and finger choices. However, future work could work conversely as well, i.e., given the placement choice of sensors and the count, the tool will recommend the best gestures that can be detected. The current layout selections are measured by classification performance, but other factors like the required amount of training data, battery performance, hardware cost or dimensions of the sensing device could be integrated into future versions of the tool. It is also possible that the suggestions prove useful beyond their application with IMU data; other approaches making use of high-dimensional data from different sensors (e.g., EMG/FSR [12, 74, 178]) could potentially expand upon the suggested layouts.

The presented computational design tool's output can also be seen not as a final choice, but as a 'best guess' for further refinement. For instance, if a layout with multiple IMUs is selected, an inverse kinematics (IK) model could be applied post-hoc to the set of suggested layouts to further leverage the inherent co-activation between the fingers and refine the final layout. Analogously, the current version of the tool comprises F1 score as evaluation criteria, but does not cover other metrics. In cases where robustness against false activation is a key design concern, individually showing precision/recall scores might be beneficial. Likewise, while the tool's design is relatively easy to use, visually depicting the gestures to instruct new users and offering strategies for an alternative representation of the confusion matrix and the F1 score can aid understanding of the classification results. Inspired by Ashbrook et al. [179] and Kohlsdorf et al. [60], future versions of the tool may also incorporate techniques to estimate the chances of false positives for each gesture by comparing the selected gestures to a large corpus of everyday activity data. The benefits of this feature will build trust in the adoption of gesture recognition in real-world deployments.

In summary, the contributions in this thesis have pushed the boundaries for providing always-available input. The findings will serve as a conceptual and technical foundation that can be used to realize the vision of access to computers - wherever and whenever users need them, without interfering with their everyday lives.

List of Figures

List of Tables

References

[1] Christian Loclair, Sean Gustafson, and Patrick Baudisch. PinchWatch: a wearable device for one-handed microinteractions. In *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '10)*.

[2] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. URL https://doi.org/10.1145/2380116.2380139.

[3] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. URL https://doi.org/10.1145/2807442.2807450.

[4] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. User Elicitation on Single-Hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858589.

[5] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction (INTERACT '11)*. URL https://doi.org/10.1007/978-3-642-23774-4_45.

[6] Leonardo Angelini, Francesco Carrino, Stefano Carrino, Maurizio Caon, Omar Abou Khaled, Jürgen Baumgartner, Andreas Sonderegger, Denis Lalanne, and Elena Mugellini. Gesturing on the Steering Wheel: A User-Elicited Taxonomy. In *Proceedings of the 6th International Conference on Automotive User Interfaces and*

Interactive Vehicular Applications (AutomotiveUI '14). URL https://doi.org/10.1145/2667317.2667414.

[7] Jaime Ruiz and Yang Li. DoubleFlip: a motion gesture delimiter for mobile interaction. In *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems (CHI '11)*. URL https://doi.org/10.1145/1978942.1979341.

[8] Philip Quinn, Seungyon Claire Lee, Melissa Barnhart, and Shumin Zhai. Active Edge: Designing Squeeze Gestures for the Google Pixel 2. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300504.

[9] Hao Lu and Yang Li. Gesture On: Enabling Always-On Touch Gestures for Fast Mobile Access from the Device Standby Mode. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems (CHI '15)*. URL https: //doi.org/10.1145/2702123.2702610.

[10] Marcos Serrano, Eric Lecolinet, and Yves Guiard. Bezel-Tap Gestures: Quick Activation of Commands from Sleep Mode on Tablets. In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems (CHI '13)*. URL https: //doi.org/10.1145/2470654.2481421.

[11] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the 2005 CHI Conference on Human Factors in Computing Systems (CHI '05)*. URL https://doi.org/10.1145/1054972.1055035.

[12] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. Enabling Always-Available Input with Muscle-Computer Interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. URL https://doi.org/10.1145/1622176.1622208.

[13] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. Real-Time Pose and Shape Reconstruction of Two Interacting Hands with a Single Depth Camera. *ACM Transactions on Graphics (TOG '19)*. URL https://doi.org/10.1145/3306346.3322958.

[14] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D. Twigg, and Kenrick Kin. Online Optical Marker-Based Hand Tracking with Deep Labels. *ACM Transactions on Graphics (TOG '18)*. URL https://doi.org/10.1145/3197517.3201399.

[15] G. Schlesinger. Der mechanische Aufbau der künstlichen Glieder. In M. Borchardt, Konrad Hartmann, Leymann, Richard Radike, Schlesinger, and Schwiening, editors, *Ersatzglieder und Arbeitshilfen: Für Kriegsbeschädigte und Unfallverletzte*. URL https://doi.org/10.1007/978-3-662-33009-8_13.

[16] Robert J Schwarz. The anatomy and mechanics of the human hand. *Artificial Limbs '55*. URL https://pubmed.ncbi.nlm.nih.gov/13249858/.

[17] JL Pons, E Rocon, R Ceres, Dominiek Reynaerts, B Saro, S Levin, and W Van Moorleghem. The MANUS-HAND Dextrous Robotics Upper Limb Prosthesis: Mechanical and Manipulation Aspects. *Autonomous Robots '04*. URL https: //doi.org/10.1023/B:AURO.0000016862.38337.f1.

[18] Thea Iberall. Human Prehension and Dexterous Robot Hands. *The International Journal of Robotics Research (IJRR '97)*. URL https://doi.org/10.1177/027836499701600302.

[19] Guido Heumer, Heni Ben Amor, and Bernhard Jung. Grasp Recognition for Uncalibrated Data Gloves: A Machine Learning Approach. *Presence: Teleoperators and Virtual Environments '08*. URL https://doi.org/10.1162/pres.17.2.121.

[20] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. Grasping Microgestures: Eliciting Single-Hand Microgestures for Handheld Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https: //doi.org/10.1145/3290605.3300632.

[21] Adwait Sharma, Michael A. Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. SoloFinger: Robust Microgestures While Grasping Everyday Objects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. URL https://doi.org/10.1145/3411764.3445197.

[22] Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-Grained Finger Microgestures. *ACM Transactions on Computer-Human Interaction (TOCHI '22)*. URL https://doi.org/10.1145/3569894.

[23] Seongkook Heo, Michelle Annett, Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. No Need to Stop What You'Re Doing: Exploring No-Handed Smartwatch Interaction. In *Proceedings of the Graphics Interface Conference (GI '17)*. URL https://dl.acm.org/doi/10.5555/3141475.3141498.

[24] Gierad Laput and Chris Harrison. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300568.

[25] Alessandro Carfì, Timothy Patten, Yingyi Kuang, Ali Hammoud, Mohamad Alameh, Elisa Maiettini, Abraham Itzhak Weinberg, Diego Faria, Fulvio Mastrogiovanni, Guillem Alenyà, Lorenzo Natale, Véronique Perdereau, Markus Vincze, and Aude Billard. Hand-Object Interaction: From Human Demonstrations to Robot Manipulation. *Frontiers in Robotics and AI '21*. URL https://www.frontiersin.org/article/10.3389/frobt.2021.714023.

[26] John R Napier. The prehensile movements of the human hand. *The Journal of Bone and Joint Surgery. British volume '56*. URL https://doi.org/10.1302/0301-620X.38B4.902.

[27] Mark R Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation '89*. URL https://doi.org/10.1109/70.34763.

[28] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. The GRASP Taxonomy of Human Grasp Types. *IEEE Transactions on Human-Machine Systems '16*. URL https://doi.org/10.1109/thms.2015.2470657.

[29] Christine L MacKenzie and Thea Iberall. *The Grasping Hand: Volume 104 (Advances in Psychology)*.

[30] Lian Loke and Toni Robertson. Moving and Making Strange: An Embodied Approach to Movement-Based Interaction Design. *ACM Transactions on Computer-Human Interaction (TOCHI '13)*. URL https://doi.org/10.1145/2442106.2442113.

[31] Maiken Hillerup Fogtmann, Jonas Fritsch, and Karen Johanne Kortbek. Kinesthetic Interaction: Revealing the Bodily Potential in Interaction Design. In *Proceedings of the 20th Australasian Conference on Computer-Human Interaction: Designing for Habitus and Habitat (OZCHI '08)*. URL https://doi.org/10.1145/1517744.1517770.

[32] Xuhai Xu, Jiahao Li, Tianyi Yuan, Liang He, Xin Liu, Yukang Yan, Yuntao Wang, Yuanchun Shi, Jennifer Mankoff, and Anind K Dey. HulaMove: Using Commodity IMU for Waist Interaction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. URL https://doi.org/10.1145/3411764.3445182.

[33] Sara Eriksson, Åsa Unander-Scharin, Vincent Trichon, Carl Unander-Scharin, Hedvig Kjellström, and Kristina Höök. Dancing With Drones: Crafting Novel Artistic Expressions Through Intercorporeality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300847.

[34] Robert Walter, Gilles Bailly, Nina Valkanova, and Jörg Müller. Cuenesics: Using Mid-Air Gestures to Select Items on Interactive Public Displays. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '14)*. URL https://doi.org/10.1145/2628363.2628368.

[35] Panagiotis Vogiatzidakis and Panayiotis Koutsabasis. 'Address and command': Two-handed mid-air interactions with multiple home devices. *International Journal of Human-Computer Studies (IJHCS '22)*. URL https://doi.org/10.1016/j.ijhcs.2021.102755.

[36] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. Experimental Analysis of Barehand Mid-Air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300426.

[37] Daniel Lee Ashbrook. *Enabling mobile microinteractions*. PhD thesis, Georgia Institute of Technology, 2010. URL http://hdl.handle.net/1853/33986.

[38] Roger Boldu, Alexandru Dancu, Denys J.C. Matthies, Pablo Gallego Cascón, Shanaka Ransir, and Suranga Nanayakkara. Thumb-In-Motion: Evaluating Thumb-to-Ring Microgestures for Athletic Activity. In *Proceedings of the Symposium on Spatial User Interaction (SUI '18)*. URL https://doi.org/10.1145/3267782.3267796.

[39] Guangchuan Li, David Rempel, Yue Liu, Weitao Song, and Carisa Harris Adamson. Design of 3D Microgestures for Commands in Virtual Reality or Augmented Reality. *Applied Sciences '21*. URL https://www.mdpi.com/2076-3417/11/14/6375.

[40] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-Defined Gestures for Surface Computing. In *Proceedings of the 2009 CHI Conference on Human Factors in Computing Systems (CHI '09)*. URL https://doi.org/10.1145/1518701.1518866.

[41] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. Maximizing the Guessability of Symbolic Input. In *Extended Abstracts of the 2005 CHI Conference on Human Factors in Computing Systems (CHI EA '05)*. URL http://doi.acm.org/10.1145/1056808.1057043.

[42] Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. Understanding Users' Preferences for Surface Gestures. In *Proceedings of Graphics Interface (GI '10)*. URL http://dl.acm.org/citation.cfm?id=1839214.1839260.

[43] Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum. A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI. In *Gesture-Based Communication in Human-Computer Interaction (GW '03)*. URL http://dx.doi.org/10.1007/978-3-540-24598-8_38.

[44] Miguel A Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of Pre-Designed and User-Defined Gesture Sets. In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems (CHI '13)*. URL https://doi.org/10.1145/2470654.2466142.

[45] Martin Weigel, Vikram Mehta, and Jürgen Steimle. More than touch: Understanding how people use skin as an input surface for mobile computing. In *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems (CHI '14)*. URL https://doi.org/10.1145/2556288.2557239.

[46] Jane L. E, Ilene L. E, James A. Landay, and Jessica R. Cauchard. Drone and Wo: Cultural Influences on Human-Drone Interaction Techniques. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. URL http://doi.acm.org/10.1145/3025453.3025755.

[47] Tanja Döring, Dagmar Kern, Paul Marshall, Max Pfeiffer, Johannes Schöning, Volker Gruhn, and Albrecht Schmidt. Gestural interaction on the steering wheel. In *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems (CHI '11)*. URL https://doi.org/10.1145/1978942.1979010.

[48] Yanke Tan, Sang Ho Yoon, and Karthik Ramani. BikeGesture: User Elicitation and Performance of Micro Hand Gesture as Input for Cycling. In *Extended Abstracts of the 2017 CHI Conference on Human Factors in Computing Systems (CHI EA '17)*. URL https://doi.org/10.1145/3027063.3053075.

[49] Sang-Su Lee, Sohyun Kim, Bopil Jin, Eunji Choi, Boa Kim, Xu Jia, Daeeop Kim, and Kun-pyo Lee. How Users Manipulate Deformable Displays as Input Devices. In *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems (CHI '10)*. URL https://doi.org/10.1145/1753326.1753572.

[50] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the 1995 CHI Conference on Human Factors in Computing Systems (CHI '95)*. URL https: //doi.org/10.1145/223904.223964.

[51] Brandon T. Taylor and V. Michael Bove. Graspables: Grasp-Recognition as a User Interface. In *Proceedings of the 2009 CHI Conference on Human Factors in Computing Systems (CHI '09)*. URL https://doi.org/10.1145/1518701.1518842.

[52] Daniel Wigdor and Dennis Wixon. Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. 2011.

[53] Frederic Kerber, Philipp Schardt, and Markus Löchtefeld. WristRotate: a personalized motion gesture delimiter for wrist-worn devices. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia (MUM '15)*. URL https://doi.org/10.1145/2836041.2836063.

[54] Bryan Wang and Tovi Grossman. BlyncSync: Enabling Multimodal Smartwatch Gestures with Synchronous Touch and Blink. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. URL https: //doi.org/10.1145/3313831.3376132.

[55] Euan Freeman, Gareth Griffiths, and Stephen A. Brewster. Rhythmic micro-gestures: discreet interaction on-the-go. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI '17)*. URL https://doi.org/10.1145/3136755.3136815.

[56] Huy Viet Le, Sven Mayer, Benedict Steuerlein, and Niels Henze. Investigating Unintended Inputs for One-Handed Touch Interaction Beyond the Touchscreen. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. URL https://doi.org/10.1145/3338286.3340145.

[57] Gierad Laput, Robert Xiao, and Chris Harrison. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. URL https://doi.org/10.1145/2984511.2984582.

[58] Travis Deyle, Szabolcs Palinko, Erika Shehan Poole, and Thad Starner. Hambone: A bio-acoustic gesture interface. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC '07)*. URL https://doi.org/10.1109/ISWC.2007.4373768.

[59] Ryo Kawahata, Atsushi Shimada, Takayoshi Yamashita, Hideaki Uchiyama, and Rin ichiro Taniguchi. Design of a Low-false-positive Gesture for a Wearable Device. In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM '16)*. URL https://doi.org/10.5220/0005701905810588.

[60] Daniel Kohlsdorf, Thad Starner, and Daniel Ashbrook. MAGIC 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG '11)*. URL https://doi.org/10.1109/FG.2011.5771412.

[61] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 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 (UIST '16)*. URL https://doi.org/10.1145/2984511.2984565.

[62] Jun Gong, Yang Zhang, Xia Zhou, and Xing-Dong Yang. Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. URL https://doi.org/10.1145/3126594.3126615.

[63] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. Grips and gestures on a multi-touch pen. In *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems (CHI '11)*. URL https://doi.org/10.1145/1978942.1979138.

[64] Daniel Groeger and Jürgen Steimle. ObjectSkin: Augmenting Everyday Objects with Hydroprinted Touch Sensors and Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '18)*. URL https://doi.org/10.1145/3161165.

[65] Hsin-Liu Kao, Artem Dementyev, Joseph A Paradiso, and Chris Schmandt. NailO: fingernails as an input surface. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems (CHI '15)*. URL https://doi.org/10.1145/2702123.2702572.

[66] Yasha Iravantchi, Mayank Goel, and Chris Harrison. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300245.

[67] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the ACM International Conference on Interactive Surfaces and Spaces (ISS '18)*. URL https://doi.org/10.1145/3279778.3279799.

[68] Yuta Sugiura, Fumihiko Nakamura, Wataru Kawai, Takashi Kikuchi, and Maki Sugimoto. Behind the palm: Hand gesture recognition through measuring skin deformation on back of hand by using optical sensors. In *Proceedings of the 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE '17)*. URL https://doi.org/10.23919/SICE.2017.8105457.

[69] Jess McIntosh, Asier Marzo, and Mike Fraser. SensIR: Detecting Hand Gestures with a Wearable Bracelet Using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. URL https://doi.org/10.1145/3126594.3126604.

[70] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. URL https: //doi.org/10.1145/3025453.3025807.

[71] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. URL https://doi.org/10.1145/3332165.3347867.

[72] Daniele Esposito, Emilio Andreozzi, Gaetano D. Gargiulo, Antonio Fratini, Giovanni D'Addio, Ganesh R. Naik, and Paolo Bifulco. A Piezoresistive Array Armband With Reduced Number of Sensors for Hand Gesture Recognition. *Frontiers in Neurorobotics '20*. URL https://doi.org/10.3389/fnbot.2019.00114.

[73] Martin Weigel and Jürgen Steimle. DeformWear: Deformation Input on Tiny Wearable Devices. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '17)*. URL https://doi.org/10.1145/3090093.

[74] Artem Dementyev and Joseph A. Paradiso. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. URL https://doi.org/10.1145/2642918.2647396.

[75] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. AuraRing: Precise Electromagnetic Finger Tracking. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '19)*. URL https://doi.org/10.1145/3369831.

[76] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858483.

[77] Daniel Ashbrook, Patrick Baudisch, and Sean White. Nenya: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring. In *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems (CHI '11)*. URL https://doi.org/10.1145/1978942.1979238.

[78] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. TouchRing: Subtle and Always-Available Input Using a Multi-Touch Ring. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. URL https://doi.org/10.1145/2957265.2961860.

[79] Hsin-Liu Cindy Kao, Abdelkareem Bedri, and Kent Lyons. SkinWire: Fabricating a Self-Contained On-Skin PCB for the Hand. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '18)*. URL https://doi.org/10.1145/3264926.

[80] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Interactive Hand Pose Estimation Using a Stretch-Sensing Soft Glove. *ACM Transactions on Graphics (TOG '19)*. URL https://doi.org/10.1145/3306346.3322957.

[81] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature '19*. URL https://doi.org/10.1038/s41586-019-1234-z.

[82] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. A Survey on Hand Pose Estimation with Wearable Sensors and Computer-Vision-Based Methods. *Sensors '20*. URL https://doi.org/10.3390/s20041074.

[83] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. ThumbRing: Private Interactions Using One-Handed Thumb Motion Input on Finger Segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. URL https://doi.org/10.1145/2957265.2961859.

[84] Ju Young Oh, Ji-Hyung Park, and Jung-Min Park. FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality. *IEEE Access '20*. URL https://doi.org/10.1109/ACCESS.2020.2997972.

[85] Katrin Wolf, Robert Schleicher, Sven Kratz, and Michael Rohs. Tickle: A Surface-Independent Interaction Technique for Grasp Interfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. URL https://doi.org/10.1145/2460625.2460654.

[86] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '21)*. URL https://doi.org/10.1145/3478114.

[87] Sandra Bardot, Surya Rawat, Duy Thai Nguyen, Sawyer Rempel, Huizhe Zheng, Bradley Rey, Jun Li, Kevin Fan, Da-Yuan Huang, Wei Li, and Pourang Irani. ARO: Exploring the Design of Smart-Ring Interactions for Encumbered Hands. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI '21)*. URL https://doi.org/10.1145/3447526.3472037.

[88] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. URL https: //doi.org/10.1145/3332165.3347947.

[89] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '20)*. URL https://doi.org/10.1145/3397309.

[90] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. BackHand: Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. URL https://doi.org/10.1145/2807442.2807462.

[91] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. URL https://doi.org/10.1145/3338286.3340147.

[92] Matthias Schröder, Jonathan Maycock, and Mario Botsch. Reduced Marker Layouts for Optical Motion Capture of Hands. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG '15)*. URL https://doi.org/10.1145/2822013.2822026.

[93] Sheldon Andrews, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell. Real-Time Physics-Based Motion Capture with Sparse Sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP '16)*. URL https://doi.org/10.1145/2998559.2998564.

[94] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *ACM Transactions on Graphics (TOG '18)*. URL https://doi.org/10.1145/3272127.3275108.

[95] Karsten Eckhoff, Manon Kok, Sergio Lucia, and Thomas Seel. Sparse Magnetometer-free Inertial Motion Tracking – A Condition for Observability in Double Hinge Joint Systems. *21st IFAC World Congress '20*. URL https://doi.org/10.1016/j.ifacol.2020.12.403.

[96] Bingni W Brunton, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Optimal sensor placement and enhanced sparsity for classification. *arXiv preprint*. URL https://doi.org/10.48550/arXiv.1310.4217.

[97] Stergios Poularakis, Grigorios Tsagkatakis, Panagiotis Tsakalides, and Ioannis Katsavounidis. Sparse representations for hand gesture recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '13)*. URL https://doi.org/10.1109/ICASSP.2013.6638358.

[98] Tomás Mantecón, Ana Mantecón, Carlos R. del Blanco, Fernando Jaureguizar, and Narciso García. Enhanced gesture-based human-computer interaction through a Compressive Sensing reduction scheme of very large and efficient depth feature descriptors. In *12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '15)*. URL https://doi.ieeecomputersociety.org/10.1109/AVSS.2015.7301804.

[99] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. URL https://doi.org/10.1145/1294211.1294238.

[100] Allan Christian Long, James A Landay, and Lawrence A Rowe. *Quill: a gesture design tool for pen-based user interfaces*. 2001.

[101] Ju-Whan Kim and Tek-Jin Nam. EventHurdle: Supporting Designers' Exploratory Interaction Prototyping with Gesture-Based Sensors. In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems (CHI '13)*. URL https: //doi.org/10.1145/2470654.2470691.

[102] Ju-Whan Kim, Han-Jong Kim, and Tek-Jin Nam. M.Gesture: An Acceleration-Based Gesture Authoring System on Multiple Handheld and Wearable Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858358.

[103] Aman Parnami, Apurva Gupta, Gabriel Reyes, Ramik Sadana, Yang Li, and Gregory D. Abowd. Mogeste: A Mobile Tool for In-Situ Motion Gesture Design. In *Proceedings of the 8th Indian Conference on Human Computer Interaction (IHCI '16)*. URL https://doi.org/10.1145/3014362.3014365.

[104] Hao Lü and Yang Li. Gesture Coder: A Tool for Programming Multi-Touch Gestures by Demonstration. In *Proceedings of the 2012 CHI Conference on Human Factors in Computing Systems (CHI '12)*. URL https://doi.org/10.1145/2207676.2208693.

[105] G. Holmes, A. Donkin, and I.H. Witten. WEKA: a machine learning workbench. In *Proceedings of Australian New Zealand Intelligent Information Systems Conference (ANZIIS '94)*. URL https://doi.org/10.1109/ANZIIS.1994.396988.

[106] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research '11*. URL https://doi.org/10.48550/arXiv.1201.0490.

[107] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Amy J. Ko, and James Landay. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. URL https://doi.org/10.1145/1866029.1866038.

[108] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. URL https://doi.org/10.1145/3334480.3382839.

[109] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. URL https://doi.org/10.1145/3025453.3025739.

[110] Jia Liu, Fangxiaoyu Feng, Yuzuko C Nakamura, and Nancy S Pollard. A taxonomy of everyday grasps in action. In *IEEE-RAS International Conference on Humanoid Robots '14*. URL https://doi.org/10.1109/HUMANOIDS.2014.7041420.

[111] Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, m. c. schraefel, and Jacob O. Wobbrock. Reducing Legacy Bias in Gesture Elicitation Studies. *Interactions '14*. URL http://doi.acm.org/10.1145/2591689.

[112] William A. S. Buxton. Human-computer interaction: toward the year 2000. chapter Chunking and Phrasing and the Design of Human-computer Dialogues. 1995. URL http://dl.acm.org/citation.cfm?id=212925.212970.

[113] Radu-Daniel Vatavu and Jacob O. Wobbrock. Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems (CHI '15)*. URL http://doi.acm.org/10.1145/2702123.2702223.

[114] Patryk Pomykalski, Mikołaj P. Woźniak, Paweł W. Woźniak, Krzysztof Grudzień, Shengdong Zhao, and Andrzej Romanowski. Considering Wake Gestures for Smart Assistant Use. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. URL https://doi.org/10.1145/3334480.3383089.

[115] Vanessa Cobus and Wilko Heuten. To Beep or Not to Beep? Evaluating Modalities for Multimodal ICU Alarms. *Multimodal Technologies and Interaction '19*. URL https://doi.org/10.3390/mti3010015.

[116] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '18)*. URL https://doi.org/10.48550/arXiv.1704.02463.

[117] Victor Gonzalez-Sanchez, Jennifer Rowson, and Alaster Yoxall. Analysis of finger movement coordination during the Variable Dexterity Test and comparative activities of daily living. *International Journal of Therapy and Rehabilitation '16*. URL https://doi.org/10.12968/ijtr.2016.23.10.481.

[118] John Williamson. Fingers of a Hand Oscillate Together: Phase Syncronisation of Tremor in Hover Touch Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858235.

[119] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '17)*. URL https://doi.org/10.1109/cvpr.2017.279.

[120] OptiTrack. Motive - Optical motion capture software, (accessed August 10, 2020). URL https://optitrack.com/products/motive/.

[121] Marc H Schieber. Individuated finger movements of rhesus monkeys: a means of quantifying the independence of the digits. *Journal of Neurophysiology '91*. URL https://doi.org/10.1152/jn.1991.65.6.1381.

[122] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems (CHI '15)*. URL https://doi.org/10.1145/2702123.2702136.

[123] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM '18*. URL https://doi.org/10.1145/3233231.

[124] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *International Conference on Data Science and Advanced Analytics (DSAAI '18)*. URL https://doi.org/10.1109/dsaa.2018.00018.

[125] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. URL https://doi.org/10.1145/3313831.3376219.

[126] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300233.

[127] Katrin Wolf, Sven Mayer, and Stephan Meyer. Microgesture Detection for Remote Interaction with Mobile Devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. URL https://doi.org/10.1145/2957265.2961865.

[128] Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, (accessed August 10, 2020). URL https://github.com/fmfn/BayesianOptimization.

[129] Ian M. Bullock, Joshua Z. Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M. Dollar. Grasp Frequency and Usage in Daily Household and Machine Shop Tasks. *IEEE Transactions on Haptics '13*. URL https://doi.org/10.1109/TOH.2013.6.

[130] Noitom. Hi5 VR GLOVE, (accessed August 10, 2020). URL https://hi5vrglove.com.

[131] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing '18*. URL https://doi.org/10.1016/j.neucom.2018.03.067.

[132] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters '19*. URL https://doi.org/10.1016/j.patrec.2018.02.010.

[133] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '15)*. URL https://doi.org/10.1109/CVPR.2015.7298941.

[134] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '18)*. URL https://doi.org/10.48550/arXiv.1712.01057.

[135] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified Egocentric Recognition of 3D Hand-Object Poses and Interactions. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '19)*. URL https://doi.org/10.1109/cvpr.2019.00464.

[136] Narjes Pourjafarian, Anusha Withana, Joseph A. Paradiso, and Jürgen Steimle. Multi-Touch Kit: A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing

Using a Commodity Microcontroller. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. URL https://doi.org/10.1145/3332165.3347895.

[137] Brandon T. Taylor and V Michael Bove. The Bar of Soap: A Grasp Recognition System Implemented in a Multi-functional Handheld Device. In *Extended Abstracts of the 2008 CHI Conference on Human Factors in Computing Systems (CHI EA '08)*. URL https://doi.org/10.1145/1358628.1358874.

[138] Kazuyuki Arimatsu and Hideki Mori. Evaluation of Machine Learning Techniques for Hand Pose Estimation on Handheld Device with Proximity Sensor. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. URL https://doi.org/10.1145/3313831.3376712.

[139] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. URL https://doi.org/10.1145/2501988.2502016.

[140] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Transactions on Graphics (TOG '16)*. URL https://doi.org/10.1145/2897824.2925953.

[141] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. How We Type: Movement Strategies and Performance in Everyday Typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858233.

[142] Huy Viet Le, Sven Mayer, and Niels Henze. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. URL https://doi.org/10.1145/3242587.3242605.

[143] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *Proceedings of European Conference on Computer Vision (ECCV '20)*. URL https://doi.org/10.48550/arxiv.2007.09545.

[144] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *Proceedings of European*

Conference on Computer Vision (ECCV '20). URL https://doi.org/10.1007/978-3-030-58548-8_34.

[145] Manus | Prime II for Mocap, (accessed March 13, 2021). URL https://www.manus-vr.com/mocap-gloves.

[146] Christina Salchow-Hömmen, Leonie Callies, Daniel Laidig, Markus Valtin, Thomas Schauer, and Thomas Seel. A Tangible Solution for Hand Motion Tracking in Clinical Applications. *Sensors '19*. URL https://doi.org/10.3390/s19010208.

[147] Markus Valtin, Christina Salchow-Hömmen, Thomas Seel, Daniel Laidig, and Thomas Schauer. Modular finger and hand motion capturing system based on inertial and magnetic sensors. *Current Directions in Biomedical Engineering '17*. URL https://doi.org/10.1515/cdbme-2017-0005.

[148] 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. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. URL https://doi.org/10.1145/3491102.3501904.

[149] L Jäncke, H Steinmetz, S Benilow, and U Ziemann. Slowing fastest finger movements of the dominant hand with low-frequency rTMS of the hand area of the primary motor cortex. *Experimental Brain Research '04*. URL https://doi.org/10.1007/s00221-003-1719-7.

[150] James N Ingram, Konrad P Körding, Ian S Howard, and Daniel M Wolpert. The statistics of natural hand movements. *Experimental Brain Research '08*. URL https://doi.org/10.1007/s00221-008-1355-3.

[151] David F. Fouhey, Wei-cheng Kuo, Alexei A. Efros, and Jitendra Malik. From Lifestyle Vlogs to Everyday Interactions. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '18)*. URL https://doi.org/10.48550/arXiv.1712.02310.

[152] Junjun Fan, Xiangmin Fan, Feng Tian, Yang Li, Zitao Liu, Wei Sun, and Hongan Wang. What is That in Your Hand? Recognizing Grasped Objects via Forearm Electromyography Sensing. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '18)*. URL https://doi.org/10.1145/3287039.

[153] Sijiang Du and M. Vuskovic. Temporal vs. spectral approach to feature extraction from prehensile EMG signals. In *Proceedings of the International Conference on*

Information Reuse and Integration (IRI '04). URL https://doi.org/10.1109/IRI.2004.1431485.

[154] Lynette A Jones and Susan J Lederman. *Human hand function*. 2006. URL https://doi.org/10.1093/acprof:oso/9780195173154.001.0001.

[155] Lefan Wang, Turgut Meydan, and Paul Ieuan Williams. A Two-Axis Goniometric Sensor for Tracking Finger Motion. *Sensors '17*. URL https://doi.org/10.3390/s17040770.

[156] Alan Poston. Human engineering design data digest. *Washington, DC: Department of Defense Human Factors Engineering Technical Advisory Group*, 2000.

[157] Aakar Gupta, Jiushan Yang, and Ravin Balakrishnan. Asterisk and Obelisk: Motion Codes for Passive Tagging. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. URL https://doi.org/10.1145/3242587.3242637.

[158] Hui-Shyong Yeo, Juyoung Lee, Hyung-il Kim, Aakar Gupta, Andrea Bianchi, Daniel Vogel, Hideki Koike, Woontack Woo, and Aaron Quigley. WRIST: Watch-Ring Interaction and Sensing Technique for Wrist Gestures and Macro-Micro Pointing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. URL https://doi.org/10.1145/3338286.3340130.

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

[160] Stephen Shiao-ru Lin, Nisal Menuka Gamage, Kithmini Herath, and Anusha Withana. MyoSpring: 3D Printing Mechanomyographic Sensors for Subtle Finger Gesture Recognition. In *Proceedings of the 16th International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. URL https://doi.org/10.1145/3490149.3501321.

[161] Hyejoo Kim, Hyeon-Joo Kim, Jinyoon Park, Jeh-Kwang Ryu, and Seung-Chan Kim. Recognition of Fine-Grained Walking Patterns Using a Smartwatch with Deep Attentive Neural Networks. *Sensors '21*. URL https://doi.org/10.3390/s21196393.

[162] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O. Riedl. Friend, Collaborator,

Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. URL https://doi.org/10.1145/3290605.3300854.

[163] Gilles Louppe. Understanding random forests: From theory to practice. *arXiv preprint*. URL https://arxiv.org/abs/1407.7502.

[164] Leo Breiman. Random forests. *Machine learning '01*. URL https://doi.org/10.1023/A:1010933404324.

[165] Thomas Kosch, Kevin Wennrich, Daniel Topp, Marcel Muntzinger, and Albrecht Schmidt. The Digital Cooking Coach: Using Visual and Auditory in-Situ Instructions to Assist Cognitively Impaired during Cooking. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '19)*. URL https://doi.org/10.1145/3316782.3321524.

[166] Lea Schönherr, Maximilian Golla, Thorsten Eisenhofer, Jan Wiele, Dorothea Kolossa, and Thorsten Holz. Unacceptable, where is my privacy? Exploring Accidental Triggers of Smart Speakers. *arXiv preprint*. URL https://arxiv.org/abs/2008.00508.

[167] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. URL https://doi.org/10.1145/3332165.3347950.

[168] Xuhai Xu, Haitian Shi, Xin Yi, WenJia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K. Dey. EarBuddy: Enabling On-Face Interaction via Wireless Earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. URL https://doi.org/10.1145/3313831.3376836.

[169] Francis Kinder. Wearable VR Glove Trailer, (accessed March 21, 2021). URL https://vimeo.com/527250099.

[170] Kunpeng Huang, Ruojia Sun, Ximeng Zhang, Md. Tahmidul Islam Molla, Margaret Dunne, Francois Guimbretiere, and Cindy Hsin-Liu Kao. WovenProbe: Probing Possibilities for Weaving Fully-Integrated On-Skin Systems Deployable in the Field. In *Proceedings of Designing Interactive Systems Conference (DIS '21)*. URL https://doi.org/10.1145/3461778.3462105.

[171] Artem Dementyev, Tomás Vega Gálvez, and Alex Olwal. SensorSnaps: Integrating Wireless Sensor Nodes into Fabric Snap Fasteners for Textile Interfaces. In *Proceedings*

of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). URL https://doi.org/10.1145/3332165.3347913.

[172] Watch your tech: Accidental 911 dials are on the rise, (accessed April 10, 2022). URL https://spectrumlocalnews.com/tx/austin/news/2021/07/07/watch-your-tech--accidental-911-dials-are-on-the-rise#.

[173] Mark Weiser. The Computer for the 21st Century. *Mobile Computing and Communications Review (SIGMOBILE '99)*. URL https://doi.org/10.1145/329124.329126.

[174] Raphael Wimmer. Grasp Sensing for Human-Computer Interaction. In *Proceedings of the 5th International Conference on Tangible, Embedded, and Embodied Interaction (TEI '10)*. URL https://doi.org/10.1145/1935701.1935745.

[175] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE '21*. URL https://doi.org/10.1109/JPROC.2020.3004555.

[176] Juyoung Lee, Shaurye Aggarwal, Jason Wu, Thad Starner, and Woontack Woo. SelfSync: Exploring Self-Synchronous Body-Based Hotword Gestures for Initiating Interaction. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. URL https://doi.org/10.1145/3341163.3347745.

[177] Daniel Weber, Clemens Gühmann, and Thomas Seel. RIANN—A Robust Neural Network Outperforms Attitude Estimation Filters. *AI '21*. URL https://doi.org/10.3390/ai2030028.

[178] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. URL https://doi.org/10.1145/2858036.2858093.

[179] Daniel Ashbrook and Thad Starner. MAGIC: A Motion Gesture Design Tool. In *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems (CHI '10)*. URL https://doi.org/10.1145/1753326.1753653.