Sensitive and Makeable Computational Materials for the Creation of Smart Everyday Objects

Te-yen Wu Mr

Follow this and additional works at: https://digitalcommons.dartmouth.edu/dissertations

Recommended Citation
Wu, Te-yen Mr, "Sensitive and Makeable Computational Materials for the Creation of Smart Everyday Objects" (2023). Dartmouth College Ph.D Dissertations. 141.
https://digitalcommons.dartmouth.edu/dissertations/141

This Thesis (Ph.D.) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Ph.D Dissertations by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
Sensitive and Makeable Computational Materials for the Creation of Smart Everyday Objects

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

in
Computer Sciences

by Te-yen Wu

Guarini School of Graduate and Advanced Studies
Dartmouth College
Hanover, New Hampshire

May 2023

Examining Committee:

Chairman _______________
Xing-Dong Yang

Member _______________
Temiloluwa Prioleau

Member _______________
Xia Zhou

Member _______________
Teddy Seyed

Member _______________
Gregory Abowd

F. Jon Kull, Ph.D.
Dean of Guarini School of Graduate and Advanced Studies
The vision of computational materials is to create smart everyday objects using the materials that have sensing and computational capabilities embedded into them. However, today’s development of computational materials is limited because its interfaces (i.e. sensors) are unable to support wide ranges of human interactions, and withstand the fabrication methods of everyday objects (e.g. cutting and assembling). These barriers hinder citizens from creating smart everyday objects using computational materials on a large scale.

To overcome the barriers, this dissertation presents the approaches to develop computational materials to be 1) sensitive to a wide variety of user interactions, including explicit interactions (e.g. user inputs) and implicit interactions (e.g. user contexts), and 2) makeable against a wide range of fabrication operations, such cutting and assembling. I exemplify the approaches through five research projects on two common materials, textile and wood. For each project, I explore how a material interface can be made to sense user inputs or activities, and how it can be optimized to balance sensitivity and fabrication complexity. I discuss the sensing algorithms and machine learning model to interpret the sensor data as high-level abstraction and interaction. I show the practical applications of developed computational materials. I demonstrate the evaluation study to validate their performance and robustness.

In the end of this dissertation, I summarize the contributions of my thesis and discuss future directions for the vision of computational materials.
Acknowledgements

I would like to express my sincere gratitude to my wife, Hsiao-Yun Hsu, who has been a constant source of support and encouragement throughout my PhD journey. Her unwavering faith in my abilities has been the bedrock of my success. I am also deeply grateful to my family for their love and support, which has been an important source of strength during challenging times.

I would also like to acknowledge my advisor, Xing-dong Yang, for his guidance, wisdom, and patience, which have been invaluable in shaping my research. I am also grateful for my PhD colleagues, Jun, Zheer, Josh, Quijie, Ruibo and Yiren, who provide insightful feedback and constructive criticism, which have helped me refine my research. I would also like to thank the graduated and undergraduate students who have been a part of my research team. Their enthusiasm and passion on the research projects have been a constant source of motivation and inspiration for me.

Finally, I want to especially honor the memory of my mother, who passed away during my PhD. Her dedication to educating and supporting me have been the driving force behind my pursuit of higher education. I am deeply grateful for the values she instilled in me and for the foundation she laid, which has enabled me to achieve my academic goals.
## Contents

1 Introduction and Purpose .............................................. 1  
   1.1 Document Organization ........................................ 4  

2 Background .......................................................... 5  
   2.1 Weiser’s ubiquitous computing .................................. 5  
   2.2 Internet-of-Things ............................................... 6  
   2.3 Tools for Ubicomp ............................................... 8  
   2.4 Computational Materials ....................................... 10  
   2.5 Sensitive and Makeable Computational Materials ............... 11  

3 Sensitive Computational Material - Sensing Explicit Inputs ............. 13  
   3.1 Background ....................................................... 14  
   3.2 Chapter Overview ............................................... 15  
      3.2.1 ThreadSense ................................................. 15  
      3.2.2 Fabriccio .................................................... 16  
   3.3 ThreadSense ....................................................... 17  
      3.3.1 Introduction ............................................... 17  
      3.3.2 Related Work ............................................... 19  
      3.3.3 Sensing Principle .......................................... 21  
      3.3.4 System Design ............................................. 25  
      3.3.5 Implementation ............................................ 29  
      3.3.6 Demo Applications ......................................... 31  
      3.3.7 Evaluation ................................................. 34  
      3.3.8 Supplement Study .......................................... 40  
      3.3.9 Discussion and Future Work ................................. 43  
      3.3.10 Conclusion ................................................ 44  
   3.4 Fabriccio ........................................................ 46  
      3.4.1 Introduction ............................................... 46  
      3.4.2 Related Work ............................................... 47  
      3.4.3 Sensing Principle and Background ............................ 50  
      3.4.4 Sensor Design ............................................. 51  
      3.4.5 Implementation ............................................ 57  
      3.4.6 Applications and Scenarios ................................ 61  
      3.4.7 Evaluation ................................................. 63  
      3.4.8 Effect of Covering Fabric .................................. 68
5.2.12 Evaluation .................................................. 164
5.2.13 Limitations and Future Work ............................. 171
5.2.14 Conclusion .................................................. 173

6 Conclusion ....................................................... 174
6.1 Thesis Contributions ........................................ 174
6.2 Future Directions ............................................ 175
  6.2.1 Output ....................................................... 175
  6.2.2 Computing and Networking ............................. 175
  6.2.3 Makeability and Scalability ............................ 177
  6.2.4 Software Platform ....................................... 178
Chapter 1

Introduction and Purpose

Ubiquitous computing envisions a world where computers are integrated into everyday objects to create a smart environment that can seamlessly interpret and respond to a user’s actions and intentions [1]. However, this vision is still far from becoming a reality due to the challenges of integrating computers into everyday objects on a large scale [2, 3]. For example, today’s computers are often made of rigid, inflexible and highly technical components, such as stacks of circuit boards and stiff touchscreens, which are not well-suited to the diverse forms of everyday objects. As a result, devices consisting of these components inevitably look and feel distinctively different from everyday objects made of natural materials, which makes them obtrusive in the environment rather than blending in with their surroundings.

Fortunately, the emergence of computational materials sheds new light on this problem [4]. Devices no longer need to be made of rigid and bulky components, but instead can be created using materials that have computational capabilities embedded into them [5, 6]. These devices, made of computational materials, will look and feel like everyday objects yet still have capabilities to sense and react to a user’s inputs or intentions, which I refer to as smart everyday objects. We envision that computational materials will enable the creation of smart everyday objects on a large scale, as they provide physical affordances that
allow citizens to easily incorporate them into everyday objects with established fabrication methods (e.g., cutting and assembling). With the proliferation of smart everyday objects made of computational materials, we can finally move closer to the vision of ubiquitous computing.

While technology seems to signal a time when we will be able to have computational materials to create smart everyday objects, many challenges still remain. With respect to the interfaces of computational materials, some of the most significant challenges are related to missing knowledge on (1) how to enable computational materials to sense rich user input and their context of use; and (2) how to allow computational materials to be used with established fabrication methods to create everyday objects. For example, with today’s technology, computational materials are limited to knowing user inputs and user contexts. Without an awareness of human interactions, computational materials cannot be used to create smart everyday objects, such as a table that can recognize the objects on it and the activities that a user performs. In addition to this problem, state-of-the-art computational materials will most likely break when connected components are inserted into the material for assembly (e.g., shorted circuits caused by metallic connectors like screws and nails). Due to this limitation, makers and workers hardly use computational materials to create everyday objects. I refer to this as the makeability problem in this dissertation.

To overcome these two challenges, this dissertation presents novel approaches to create new interfaces for computational materials that are more sensitive to a wide variety of user inputs and contexts, while remaining functional against a wide range of fabrication operations commonly seen during the creation of daily objects. These approaches involve building computational materials from scratch, typically including (1) new designs of electronic and material structures (e.g., textile antennas), (2) machine simulations and experiments for structure optimization, (3) novel fabrication methods, and (4) sensing and machine learning algorithms for input recognition. This thesis exemplifies the approaches through two common materials, textile and wood and demonstrate that the proposed meth-
ods are generalizable to a wider variety of other materials.

This dissertation can be divided into two parts. Part I aims to develop and evaluate new input interfaces of computational materials to enable explicit and implicit inputs on everyday objects made of textile. Part II aims to create makeable computational plywood that allows the interfaces to remain functional through the fabrication process of smart everyday objects. Through four published papers, I demonstrate that Part I can be accomplished by introducing new sensing capabilities to the existing computational textile. Specifically, I developed textile sensors that can detect touch-less finger gestures [7, 8] and recognize objects it is in contact with [9, 10]. In this way, users can interact with smart environments using gestural inputs via smart everyday objects made of computational textile (e.g., arm rest of a sofa). Furthermore, the computers are now aware of how textile everyday objects are being used (e.g., what is carried inside a user’s jean pocket). This enables many contextual applications that are not currently possible with existing technologies.

In Part II, I work on computational plywood because plywood is also a common material widely used to create everyday objects such as home items and building infrastructure [11]. Developing computational plywood could largely complement my research on computational textiles. In that project, I identify and address the makeability issues existing in computational plywood, which are also likewise the problems existing in computational textiles. Specifically, I develop a computational plywood prototype that can detect a user’s gesture inputs and activities for explicit and implicit interactions, while it is still be able to survive the woodworking operations of the tools designed for their non-computational counterparts. To validate the effectiveness of the computational plywood, I recruit a woodworker to make three everyday objects using the computational plywood prototype, including table, nightstand and cutting board. I evaluate the performance of these objects made of computational plywood to show the robustness and practicality of makeable computational plywood.

My dissertation will contribute to the HCI/Ubicomp community through: (1) identify-
ing challenges in using computational materials to create smart everyday objects; (2) new knowledge on how to make the interface of computational textile and wood more sensitive and makeable; and (3) demonstrating how smart everyday objects can be better created using sensitive and makeable computational textile and plywood.

1.1 Document Organization

Chapter 2 introduces the background of computational materials and situates my thesis in the relevant literature. Chapters 3 and 4 provide the research projects in Part I of my thesis, which focuses on introducing new sensing capabilities to computational textiles. This part can be further divided into two subtopics: (1) enhancing explicit inputs on computational materials (Chapters 3); (2) enabling implicit inputs on computational materials (Chapters 4). Part II, including Chapters 5, presents the concept of makeable computational materials and demonstrates it through a research project that enables computational plywood to be makeable. Chapter 6 concludes the thesis and summarizes potential future work inspired by this research.
Chapter 2

Background

This chapter aims to provide a comprehensive overview of the background of my dissertation. It begins by tracing the evolution of ubiquitous computing (Ubicomp) and examining the challenges it has encountered in large-scale deployment. I explore existing solutions and tools that may help overcome the challenges but also argue for the potential of computational materials to address these issues completely. Lastly, I review existing research on computational materials and highlight the novel contributions of my dissertation to this field of study.

2.1 Weiser’s ubiquitous computing

In the 1990s, Mark Weiser, a computer pioneer, sketched the vision of ubiquitous computing (Ubicomp) in a Scientific American article [1]. His seminal remark of "weaving themselves into the fabric of everyday life until they are indistinguishable from it" ushered computers into a new era. Computers are no longer just bulky electronic devices sitting on desks, but can be mobile devices we carry in our pockets, wearable devices on our wrists, or home appliances in our houses. This landmark article has inspired numerous computer scientists and researchers to rethink what computing technology should be created and how it should be designed. It has had significant influence on the computer world for decades.
In Weiser’s original article, ubiquitous computing refers to computers that are ubiquitous enough to vanish into the background. It is not just a consequence of technology, but is more about human psychology. Whenever people learn technology sufficiently well, they cease to be aware of it, so the technology fades into the background of consciousness. Weiser’s group demonstrated this concept by using several examples, such as tabs, pads, and boards that can seamlessly adapt to users’ needs and behavior. They showed that as people grew accustomed to the technology and shifted their focus to their intended tasks, these devices could effectively “disappear” into the periphery [1].

In fact, with the development of technology, many devices proposed in the article have been realized and are now used in our daily lives. For example, iPads are portable devices with wide touchscreens to allow users to write and access information on them [12]. They are now widely used in schools, as students can easily access educational materials, collaborate with their peers, and complete assignments. In our daily lives, we use them without thinking, as we focus more on our goals beyond the devices. In psychology, we free to use them without thinking, as we focus more on our goals beyond the devices. In addition to iPads, smartphones have also significantly impacted our daily lives. We use them throughout the day to communicate, stay entertained, be productive, navigate, and interact with friends and family. Their powerful capabilities of context awareness (e.g., locations and activities) facilitate our use and allow us to pay more attention to our tasks. These examples all convey the message that computers have become an integrated part of our daily lives and have started to fade into the background of our consciousness.

2.2 Internet-of-Things

While computers are already integrated into our daily lives, such advancements do not stop scientists and researchers from moving forward to the next level. A broader explanation of Ubicomp was brought up, which emphasizes that computing technology will physically
disappear by being seamlessly integrated into everyday objects and environments [13]. This will involve embedding computers, sensors, and actuators into the fabric of the physical world in a natural and unobtrusive manner.

A more well-known and similar term is the Internet of Things (IoT), coined by Kevin Ashton. It is a term widely adopted by the industry and popular in markets. It describes a physical device or everyday object that has the capabilities of collecting and processing data and being connected to the internet to communicate with other devices or systems. For instance, a smartwatch is a device that embeds a computer into a traditional watch. It can be worn on the wrist to facilitate users’ interactions with computers and monitor users’ activities [14]. A smart lock is another example of an IoT device that integrates computing technology into a daily object (i.e. door lock). It allows people to use more convenient and unobtrusive ways such as footprint or face recognition to unlock the door [15]. These examples demonstrate that computers are starting to merge into our everyday objects to facilitate our daily life.

While it seems a time that we are close to the vision of ubiquitous computing, there are still many everyday objects made of natural materials (e.g. textile and wood) that are not smart. For example, the clothing we wear, and the furniture we use are yet unaware of our intentions and activities. If we look back at IoT devices and those everyday objects in the environment, we find that the devices are just a small subset of "things" and not really similar to everyday objects but sophisticated gadgets that have computing capability and wireless connectivity added. They are often made of obtrusive and rigid structural elements, such as stacks of circuit boards and touchscreens, wrapped in a fine box, which inevitably makes them look and feel distinctively different from everyday objects that are made of natural materials. Apparently, we are still far from the vision of ubiquitous computing if we interpret the second sentence, "weaving themselves into the fabric of everyday life until they are indistinguishable from it," in a more literal way.
2.3 Tools for Ubicomp

To drive the evolution of computing to the next stage, commentaries on Ubicomp have assessed what is missing and what is next for Ubicomp [2, 3, 16]. One of the major criticisms is that we lack tools and platforms to support creative designers in deploying Ubicomp technology on a large scale [3]. For example, sensing technologies are a critical component of ubiquitous computing [1]. Many sensing technologies, such as RF-based sensing [17, 18, 19], vibration-based sensing [20, 21, 22, 23], and vision-based sensing [24, 25, 26], have been proposed to enable objects (e.g., tables and walls) to interpret human interactions and activities. However, these technologies mostly require deploying sensors in the environment, on the body, or on an object, even if they could be built on existing infrastructure (e.g., Wi-Fi). Their sensing capabilities are even dependent on how many sensors are deployed, as more sensors lead to higher-resolution data for analysis. But the question is who and how to deploy these sensors seamlessly. We may expect the technology industry to take care of this, but the high barrier to technology development leads to high deployment costs and times. We cannot assume that the industry will be able to transform all diverse everyday objects around our life. We need more people, including designers, hobbyists, artists, and workers, to participate in the deployment, just as app developers proliferated applications on desktop and mobile computers. To engage them, one way is to develop hardware tools and software platforms to lower the barrier to technology deployment.

Fortunately, with the rise of maker culture and movement [27], we already have some tools and platforms to facilitate the development and deployment of Ubicomp technology. The well-known example is Arduino [28], which is an open-source electronic platform designed to make hardware development easy for hobbyists, artists, and designers to create interactive electronic projects. Besides to it, Microsoft Research’s plug-and-play platforms, such as .NET Gadgeteer [29] and Jacdac [30], are also good tools for rapid prototyping, testing, and manufacturing of custom circuit boards and housings. As an HCI an Ubicomp researcher, I have also developed tools for circuit prototyping [31, 32, 33, 34] and electronic
accessibility [35, 36] (Figure 2.1). With the effort of industry and academia, the bar to the technology is lower than ever before. Technical hobbyists are able to replicate the technology and deploy them at their home for household innovations [37, 38].

![Figure 2.1](image.png)

Figure 2.1: My research in circuit prototyping tools. (a) Circuitstack, a system that combines the flexibility of breadboarding with the correctness of printed circuits, for enabling rapid and extensible circuit construction. (b) CircuitSense, a system that automatically recognizes the wires and electronic components placed on breadboards. (c) CurrentViz, a system that can sense and visualize the electric current flowing through a circuit, which helps users quickly understand otherwise invisible circuit behavior. (d) Proxino, blending the virtual and physical worlds for prototyping circuits using physical proxies. (e) TangibleCircuit, a novel haptic and audio feedback device that allows blind and visually impaired (BVI) users to understand circuit diagrams. (f) AccessibleCircuit, the designs for low cost and 3D-printable add-on components to adapt existing breadboards, circuit components and electronics tools for blind or low vision (BLV) users

While the electronic tools and platform have facilitated the development and deployment of Ubicomp technology, the challenge to democratizing it still persists. End users such as designers and workers who are now responsible for creating everyday objects like clothing and furniture are still deterred from using these tools to build technology in their products [39]. This is because the intensive micro-electronic and programming activities involved in using these tools are challenging for ordinary citizens. For instance, using Arduino requires a user to have knowledge about voltage, current, and circuit building. To deploy Ubicomp technology in the home, the user would need to have a minimum under-
standing of sensing or computing technology to customize it to their environment [40]. This process is difficult, as end users may be out of their comfort zone and may have to work in a multidisciplinary domain on their own. As a result, only technical tinkerers are capable of using these tools to set up Ubicomp technology, while others are not yet prepared to do so. To overcome this challenge, it may be necessary to explore an entirely different approach to creating devices.

### 2.4 Computational Materials

Currently, we build devices or computers through a complicated process involving designing circuitry, building prototypes, assembling devices, and installing software. As mentioned earlier, even with the help of advanced tools, this process is still strenuous for non-technical citizens. It is because the skills and components used for a device are quite different from their daily practices of creating everyday objects. To overcome this challenge, it may be time to rethink how a computer could be made. For example, could we make a computer adapt to the fabrication process of everyday objects? One of the key terms in HCI is affordance, referring to the perceived or actual properties of an object or interface that suggest how it can be used. Applying it to today’s issue, could we create computers that have the affordance to allow non-technical users to easily use them to fabricate everyday objects? The answer is probably yes if we can make computers in the form of materials that citizens can use to create smart everyday objects with established fabrication methods.

Fortunately, advances in materials science and manufacturing provide the possibility of enabling everyday materials to have sensing, computing, and wireless connectivity capabilities "woven" into them. These capabilities are imperceptible or can look and feel like the materials into which they are embedded. Gregory Abowd coined the term "computational materials" to refer to these new and functional materials. In his paper [4], he emphasizes three dimensions that clarify the vision of computational materials: power, scale, and form.
factor. Power refers to self-sustainable computational materials that can harvest sufficient energy to carry out their computational and connectivity functions. Scale indicates the cost of manufacturing computational materials, while form factor is the property of computational materials that makes them either look and feel like everyday objects or not alter the look and feel of those objects when placed on them.

While Gregory Abowd and his colleagues initiate the vision and propose the dimensions of computational materials [4], I see the value of computational materials from a different perspective. I view computational materials as a solution to revolutionize how non-technical end-users can create a device and deploy ubiquitous computing technology. This is because computational materials have the affordance of everyday materials that citizens can easily use to create everyday objects on a large scale. In other words, the key idea is that with computational materials, non-technical users do not need to learn low-level electronics and software knowledge to create "smart" everyday objects. Instead, they can easily use computational materials to create objects that will inherently inherit computing, connectivity, and sensing capabilities from them.

2.5 Sensitive and Makeable Computational Materials

To create computational materials that can be used to build "smart" everyday objects, there are many challenges beyond power and scale. For example, how can the sensing technologies developed by the Ubicomp community also work in computational materials? Without such an understanding, objects made of computational materials are unable to receive inputs from humans. Therefore, Part I of my PhD research focuses on how to bring novel sensing capabilities to the interface of computational materials to enable explicit inputs (Chapters 3) and implicit inputs (Chapters 4). For example, I bring touch sensing [8] and touchless sensing [7] on textile materials. They could be made on cushion or sofa armrest to allow users to explicitly input to them to interact with smart environment. On top of it, I
also enable textile materials to be aware of its context, such as nearby objects the textile in contact with [10, 9]. One of the applications is that the recipe helper can better guide users to follow the steps because it can know what objects or ingredients are on the table cloth.

While the sensing capability is essential for computational materials, it is also important to consider how computational material can go through the fabrication process of everyday objects. That is one of the main reasons why we develop computational materials. To clarify this, I called the ability of computational material to remain functional against fabrication operations "makeability". Makeability is a property or affordance of regular materials. It facilitates how nontechnical users can use computational materials to deploy technologies. For example, when a woodworker uses computational wood to create a table, they do not need to learn microelectronics. Instead, they can cut it using a saw, and assemble it using screws to make a table. To exemplify this, Part 2 of my PhD research works on this aspect. I developed a piece of plywood that can sense vibration for gesture and activity recognition while still surviving common woodworking operations, such as sawing, screwing, and nailing. This plywood can be used to create furniture and artifacts [11].

In summary, I argue that computational materials are the solution to deploying Ubicomp technology on a large scale. In lieu of prior work, my thesis plays an important role in demonstrating how computational materials can be sensitive to a wide range of user inputs and contexts, and makeable against fabrication operations. In the next chapters, I will delve into the details of my research projects, and discuss future work that could be motivated by this thesis.
Chapter 3

Sensitive Computational Material - Sensing Explicit Inputs

Being a novel form of computing device, computational materials necessitate the creation of a new user interface to accommodate various forms of user inputs. One such category is explicit input, which is provided explicitly by the user. Effective explicit input methods usually draw inspiration from human communication techniques. For instance, people often use speech and pointing gestures to facilitate nuanced and expressive communication. Therefore, these techniques, such as gestures and voice commands, are considered intuitive and natural modalities for explicit input with computing devices [41]. To enable computational materials to support diverse and intuitive explicit input modalities, it is necessary to advance the sensing capabilities of these materials.

While today’s advancements in material science have shown the possibility of creating everyday materials with computing capabilities, the user interface of computational materials is still limited in sensing these rich explicit input modalities. For example, state-of-the-art computational textiles are unable to recognize mid-air gestures, as existing work mostly focuses on touch [42, 43] and deformation interactions [44, 45]. This results in missed opportunities for users to interact with objects made of textile materials through
meaningful mid-air gestures when their hands are unclean during eating. Additionally, some thin materials, such as a thread, are also unable to detect where a user touches, limiting one-dimensional (1D) touch interactions on soft, thin objects (e.g., stripes, cords) made of those materials.

In this dissertation, I demonstrate my techniques to enable textile material to sense richer explicit input through two research projects, each of which exemplifies a unique challenge and approach [7, 8]. I chose textiles in this part because it is one of the most common materials in people’s daily lives. Things that are made of textiles include clothing, furniture, decoration, toys, and bags. Further, the approaches that I developed for textiles could also be used on other materials such as paper and wood to enrich their sensing capabilities. It is because the working principles of these approaches were not specifically based on the unique properties of textiles (e.g. flexibility and stretchability), which allow them to work in other materials.

3.1 Background

Researchers have developed methods to instrument fabrics with sensors to enable a wide variety of applications through sensing touch input [42, 43], and deformation of the fabric [44, 45]. One example is the Musical Jacket, where a textile-based keypad was embroidered onto a jacket and allowed a user to touch to play music [46]. More recent examples include Project Jacquard [42] and GestureSleeve [47], both of which are capable of sensing touch gestures on different parts of a garment. For deformation-based inputs, SmartSleeve demonstrates how a textile can sense interactions such as folding, stretching and pressing [44].

In addition to interactive fabric, researchers have also explored approaches to augment papers with sensing capabilities to detect touch input [48, 49, 50], finger rubbing [49], the proximity of the hand [48], the shape of the paper [51], the deformation of the paper
For example, PrintSense introduced a new pattern of printed conductive electrode arrays on paper to support multimodal interactions including touch, pressure, the proximity of hand, and folding [48]. Shape-Aware Material used capacitive sensing to detect the shape of the paper upon cutting [51]. Paper generators leveraged the triboelectric effect to enable interactions like touch, rubbing, and sliding in a children’s storybook [49]. Also based on the triboelectric effect, SATURN demonstrated the design and fabrication of a self-powered paper microphone that can sense nearby audio [52].

While these sensing capabilities of computational materials have been well explored, there are still many under-explored (or missing) sensing capabilities, such as sensing mid-air gestural input. As a result, my research projects complement existing research in the literature by demonstrating new sensing capabilities on textiles.

3.2 Chapter Overview

In this dissertation, I present two research projects [8, 7] that enriched the explicit input modalities of computational textiles. The first project is ThreadSense [8]. It enables an interactive thread of less than 0.4 mm thick to sense one-dimensional touch input, such as touch location and gestures. The second project is Fabriccio [7]. It introduces interactive fabric that can sense midair gestures. In the following sections, I will provide an overview of these research projects to familiarize readers with my work in this area.

3.2.1 ThreadSense

In ThreadSense [8], I proposed a new sensing technique for one-dimensional touch input that works on a thread of less than 0.4 mm thick. My technique is based on impedance sensing, which can locate up to two touches with a spacing resolution that is unachievable by existing methods. My approach is unique in that it locates a touch based on a mathematical model that describes the change in thread impedance in relation to the touch locations.
This enables the system to be easily calibrated by the user touching a known location(s) on the thread, making it adaptable to various environmental settings and users.

A system evaluation showed that our system could track the slide motion of a finger with an average error distance of 6.13 mm and 4.16 mm using one and five touches for calibration, respectively. The system could also distinguish between single touch and two concurrent touches with an accuracy of 99% and could track two concurrent touches with an average error distance of 8.55 mm. I demonstrated new interactions enabled by our sensing approach in several unique applications.

### 3.2.2 Fabriccio

In Fabriccio [7], I present a touchless gesture sensing technique developed for interactive fabrics using Doppler motion sensing. My prototype was developed using a pair of loop antennas (one for transmitting and the other for receiving), made of conductive thread that was sewn onto a fabric substrate. The antenna type, configuration, transmission lines, and operating frequency were carefully chosen to balance the complexity of the fabrication process and the sensitivity of our system for touchless hand gestures, performed at a 10 cm distance. Through a ten-participant study, I evaluated the performance of our proposed sensing technique across 11 touchless gestures as well as 1 touch gesture. The study result yielded a 92.8% cross-validation accuracy and 85.2% leave-one-session-out accuracy. I conclude this project by presenting several applications to demonstrate the unique interactions enabled by my technique on soft objects.
3.3 ThreadSense

3.3.1 Introduction

Today’s innovations in computational materials [44, 53, 43, 54] are rapidly changing the way people interact with smart everyday objects. Touch input, for example, has evolved from rigid body devices like touch screens to soft, one-dimensional (1D) sensors (e.g., stripes, cords) that enable interactivity on everyday objects such as drawstrings [55, 56] and headphone wires [56, 57].

In this project, I introduce a new approach for locating up to two concurrent finger touches on a thin thread made of conductive material using the principle of impedance sensing. This approach works on cords, stripes, and more importantly, very thin threads (less than 0.4 mm thick) with a simple structure (a single line of resistive material) that cannot be supported by existing methods using pressure [43], capacitance [57], or time domain reflectometry [58]. Thus, my new approach extends touch input to everyday thin threads, such as tinsels, braids, wire crafts, or embroidery (Figure 3.1b, 3.1c, 3.1d).

![Figure 3.1: ThreadSense prototype in (a) a thin thread of less than 0.4 mm thick. My sensing technique can locate up to two-finger touches on extremely thin objects, found in (b) braided hair band, (c) embroidery, and (d) wire craft.](image)

My method is also unique in that it employs a model-based approach, which locates touches based on a mathematical model describing the change in the impedance of the thread in relation to touch locations. Unlike many of the existing techniques also using impedance sensing [53, 59, 60, 61], my system does not need to be trained. The user only
needs to perform a quick calibration by simply touching a known location(s) on the thread to inform the system of their finger impedance, needed later for locating a touch.

The simplicity of the calibration allows the system to quickly adapt to various environments and users. Environmental noises such as AC voltage can vary across environments (e.g., kitchen, cars, offices) because of different types of surrounding objects (e.g., power lines, metal objects). Such noises cause ground coupling with the thread and can affect impedance sensing. Additionally, finger impedance can vary across users because of differences in their body impedance. It can even vary for the same user with different fingers or finger conditions (e.g., wet or dry). Prior designs on impedance sensing would likely require repeated training with tedious data collection in each different settings [53]. My system, however, can adapt to these variations by only requiring the user to touch the thread one or a few times depending on the accuracy needed for an application.

I demonstrate the effectiveness of my approach through a proof-of-concept prototype (called ThreadSense) developed in the form of a thin, 3D-printed thread (Figure 3.1a). In a controlled experiment with ten participants, I tested the tracking accuracy of my sensing approach in distinguishing and locating touch positions of one versus two fingers. My study revealed that the system could sense touch location with an average error distance of 6.13 mm with a single calibration point. Increasing the number of calibration points to five reduced the error distance to 4.16 mm. The system could distinguish single touches from two-touches with an accuracy of 99% and could track two concurrent touches with an average error distance of 8.55 mm.

My contributions are: (1) a sensing technique for locating touch input on an extremely thin thread; (2) a model-based approach that enables the system to work based on a simple calibration process; (3) the result of experiments to demonstrate the effectiveness of my approaches.
3.3.2 Related Work

This work builds and extends upon prior work in a number of domains, including capacitive and impedance sensing and 1D touch input on interactive stripe, cord, and fabric.

**Touch Input on Cords and Stripes**

User input on a thin stripe or cord is mainly achieved through touch [56, 57, 62] and deformation [43, 56, 57, 62, 63]. For example, I/O braid [56] allows a user to press or twist a cable to perform input. TactileTape [64] allows the user to use the press as input. Both Cord Input [57] and Cord UIs [62] can sense touch and deformation of the cable but the resolution of touch input is limited by the number of capacitive sensing unit. With StretchEBand [65], the user can perform input by stretching a stripe. The works from Sousa, et al. [66] and Klamka, et al. [55] allow the user to perform input on a cord using a sliding bead. My work focuses on 1D touch input.

Existing 1D touch sensing techniques are mainly based on capacitive sensing [56, 57, 55, 62]. Although most of them are effective, spatial sensing resolution is limited because the wirings for a large number of electrodes needed for a high spatial resolution are extremely hard to arrange in a very thin form factor. Techniques based on time domain reflectometry (TDR) [58] need a pair of electrodes to be placed side-by-side with a gap of a certain distance (e.g., 0.5 mm in the authors’ best example), which limits how thin the thread can be made and its applications. TDR is also sensitive to the deformation of the sensor, which makes it less reliable for touch input on soft objects (e.g., hair tinsels). The techniques using the charging time of capacitance [67] can potentially locate two touch points on a single line of electrode but the technique has not been explored on a thin thread.

**Touch Input on Interactive Fabric**

In addition to stripes and cords, touch input has also been developed on soft fabrics. Existing techniques for touch input on interactive fabric can be divided into the ones based on
capacitance sensing [54] and those based on resistance sensing [43, 47, 68].

The class of work utilizing capacitive touch sensing is based on fabric capacitors made of conductive materials acting as electrode plates. On a piece of fabric, the electrodes can be created using conductive threads or inks. Musical Jacket [46], is an example of early explorations in this field. The authors used stainless-steel yarns to embroider a capacitive touch keypad on denim. A more recent work, Project Jacquard [54], describes the design and fabrication of a new type of conductive yarn that can be woven into textiles using standard looms at scale.

The approaches using resistive touch sensing area based on fabric resistors. A sensor structure commonly seen in this category of work has two conductor-layers separated by a semi-conductive middle layer. For example, GestureSleeve [47] is an interactive sleeve that allows a user to perform touch gestures on the forearm. New methods are also under development to improve the resistive sensing technology. For example, Parzer et al.’s method can reduce the complexity of the sensor using a new type of yarn comprised of a metallic thread with a resistive coating [43]. These prior works focus on sensing touch on a surface and mostly rely on a grid of threads. Instead, my work aims to sense 1D touch input with a single thread-form sensor, which is therefore not achievable with previous approaches.

**Impedance Sensing**

Electrical Impedance Tomography (EIT) has been employed in much recent work for sensing hand postures [60, 61] and 2D touch locations [53, 59, 69]. For example, Tomo [60] is a wrist-worn device that senses hand postures using EIT. The technique measures interior impedance geometry with eight electrodes inside a wristband to recognize gross hand and thumb-to-finger pinch gestures. Electrick [53] enabled touch input on a wide variety of irregular objects and surfaces using EIT. A technique based on a similar sensing principle has been used to enable the track of finger and pen movements on an interactive paper
iSoft [69] utilized EIT to track real-time continuous touch input and deformation in a flexible sensor. While most of the work mentioned above uses a single AC current frequency, Swept Frequency Capacitive Sensing (SFCS) was shown to provide richer information in sensing hand gestures on everyday objects [70] and differentiating users of a touch-screen device [71]. A more recent work, Zensei [72], used SFCS to support implicit and ubiquitous user recognition on mobile devices, furniture, and in many of the indoor environments.

In comparison to the above works, my work differs in that it provides a new approach to locating touch in an extremely thin 1D space of an interactive thread. Additionally, all the existing impedance-based touch sensing techniques require training and tedious data collection. They use machine learning to classify touch input, thus these systems would likely need to be retrained under various environments and for different users. My model-based approach replaces training with a simple calibration process, which only requires the user to touch a known location(s) on the thread. Such simple calibration allows the system to quickly adapt to various environments and users.

### 3.3.3 Sensing Principle

ThreadSense enables a thread made of evenly distributed conductive material to locate user touches. The key rationale is to sense the impact of user touches on the impedance of the thread and infer touch positions. Specifically, when each end of the thread is connected to an electrode, the impedance of the thread can be measured by inserting a small AC current between the electrodes. When a finger touches the thread, it shunts a small amount of current to ground (known as "shunt mode"), which increases the thread’s impedance measured at a certain AC frequency. More importantly, when the resistance of the thread is evenly distributed along the length of the thread, the measured impedance is dependent on the touch position on the thread. Thus, one can measure the thread’s impedance to locate the finger touch.
To further illustrate the rationale, Figure 3.2 demonstrates the equivalent circuit with and without a finger touch. Here $H$ and $L$ are the high and low voltage endpoints connecting to the electrodes, respectively. $Z_0$ is the known impedance of the thread and $Z_G$ is the impedance caused by ground coupling due to environmental noises. When a finger touches the thread at a location, described using a location coefficient $\alpha$ (expressed as a ratio from 0 to 1, with the $H$ end as 0 and $L$ end as 1), $Z_s$ is introduced by the integration of the shunting impedance of the ground ($Z_G$) and finger touch ($Z_T$), which comprises the impedance of the finger and the contact impedance of the finger touch. Note that $Z_S$ remains constant regardless of the finger touch location. Finally, the impedance of the thread segment separated by the touch location is denoted as $Z_H$ and $Z_L$.

![Figure 3.2: Equivalent circuit (a) without a finger touch and (b) with a finger touching at a location $\alpha$ on the thread.](image)

When $Z_0$ and $Z_S$ are known, the impedance of the thread $Z(\alpha)$ (with the finger at location $\alpha$) can be calculated (or estimated) as below:

$$Z(\alpha) = Z_0 + \alpha(1 - \alpha) \frac{Z_0^2}{Z_S}$$

because the circuit in Figure 3.2b can be converted into a delta network using a star-to-delta transform [73]. Since the resistance of the thread is evenly distributed, $\alpha$ represents the proportion of the resistance of the segment of the thread segment separated by the touch location. For a fixed $Z_0$ and $Z_S$, different touch location $\alpha$ results in different thread impedance, which ideally should equal to the estimation $Z(\alpha)$. This relationship forms the
basis for locating the touch position.

**Experimental Validation**

To validate the relationship between measured impedance and the estimation using the above equation, I use a pair of variable resistors to emulate the variations in $Z_H$ and $Z_L$, caused by a finger touching the sensor. Thus, the ratio of the resistance values of the variable resistors determines the location coefficient $\alpha$. I keep the summation of the resistance of them (or $Z_0$) as $1M$. As shown in Figure 3.3, the variable resistors are placed on a breadboard and connected via a copper tape representing the touch position.

![Figure 3.3: Setup of my experimental validation.](image)

Two participants were recruited in the test and were asked to press their index finger on the copper tape. I measured the resulting impedance at 80 kHz using the AD5933 impedance analyzer chip. I repeated the measurement by varying at a step size of 0.1. Figure 3.4 plots the measured impedance of the participants as varies. I fitted the data using a the least squares fit into the equation to find out $Z_S$ and plot the fitted function curve. The data fitted well with the curves, indicating that the equation is an effective estimation of the effect of on the measured impedance. Therefore, if $Z_T$, $Z_G$, and measured impedance are is known, $\alpha$ can be calculated to estimate the touch location. Figure 3.4 also shows that the curves differ between the two participants, suggesting that $Z_S$ needs to be calibrated individually.
Figure 3.4: The fitted impedance curve shown in relation to the measured impedance. The impedance increases differently under the influence of different participant’s touch, but both fit in with the equation

\[
\alpha
\]

Note that \( \alpha \) is symmetric around the center of the thread. As such, touching the left or right side of the thread is indistinguishable from each other (see Figure 3.4). This issue can be resolved by connecting the thread in series with a resistor of the same resistance (coupling resistor). Note that the coupling resistor cannot be replaced by a potentiometer as the internal structure of the potentiometer may introduce noises

Challenges

Though I have proved the feasibility of locating a single touch on a thread, challenges exist for detecting two concurrent touches and for the system to work robustly against environmental noises. Detecting multi-touch is challenging because a single touch may cause a change in the measured impedance similar to that caused by two concurrent touches at different locations. For example, at 84 kHz, the impedance measured with two touch locations at \( \alpha_1 = 0.56 \), \( \alpha_2 = 0.64 \) is close to that of a single touch with location coefficient \( \alpha = 0.7 \). Further, environmental noises may also affect the measured impedance, making it less reliable for locating touch. I present my design to address these challenges.
3.3.4 System Design

To enable the ability of multi-touch detection and enhance the system’s robustness against environmental noises, I consider a frequency sweeping approach, where thread impedance measurements are collected as I sweep the frequency of AC current. Since the thread impedance varies under different AC frequencies, frequency sweeping allows us to collect a spectrum of impedance measurements across different AC frequencies. These measurements collectively can be more robust against environmental noises and present distinguishable features between single touch and multiple touches. Frequency sweeping has been used in prior works [70] to distinguish users or recognize hand poses. I are the first to apply the technique to locate touches. Next, I discuss the model-based touch localization method and system calibration.

Model-based touch localization

I extend my discussion of sensing principle from a single AC frequency to N AC frequencies, with each frequency denoted by $f_i$, where $i = 1, 2, ..., N$. Since different AC frequencies lead to different thread impedance values, I denote the measured impedance for a single touch under $f_i$, $Z(f_i)$. Similarly, I denote the impedance estimated using the equation under $f_i$, $Z(\alpha, f_i)$. Here the problem of locating a touch position is to seek an estimated location coefficient ($\alpha^*$) that minimizes the summation of all differences between the measured and estimated impedance values, as shown below:

To search for $\alpha^*$, I used the Trust Region Reflective algorithm. As mentioned earlier, to compute $Z(\alpha, f_i)$, the system must know the corresponding $Z_G$ (the impedance caused by ground coupling from the environmental noises) and $Z_T$ (the impedance of the finger and the contact impedance caused by the finger touching the thread) at frequency $f_i$. I will discuss how this information is acquired shortly in the system calibration section.

A similar approach can be used to locate multiple concurrent touches at different locations. Here I use two touches as an example for the sake of simplicity. When a user touches
the thread using two fingers, an equivalent circuit can be described in the form shown in Figure 3.5a. This circuit can be simplified using a star-to-delta transformation, and the resulting circuit is similar to the one time transformed single touch circuit (Figure 3.5b). This step is useful as it allows us to calculate the ideal thread impedance for two touches \( Z(\alpha_1, \alpha_2, f_i) \) at different locations and AC frequencies using the method described in Sensing Principle. Finally, the estimated location coefficients \( \alpha_1^*, \alpha_2^* \) for a measured impedance spectrum can be computed as:

\[
(\alpha_1^*, \alpha_2^*) = \arg\min_{(\alpha_1, \alpha_2)} \sum_{i=1}^{N} (Z_{f_i} - Z(\alpha_1, \alpha_2, f_i))^2
\]

![Figure 3.5: Diagram of the equivalent equivalent circuit of two concurrent touches simplified into an equivalent impedance \( Z_{HL} \) circuit similar to one touch.](image)

**Distinguishing Between 1 and 2 Concurrent Touches**

Once \( \alpha^* \) and \( (\alpha^*, \alpha_2^*) \) are determined, my system is now able to tell which one is more likely the cause of the change in the measured thread impedance. In particular, for a given spectrum of measured impedance across N AC frequencies, \( Z_f \), whether the occurrence of a touch event is caused by one or two fingers can be determined by choosing between \( \alpha^* \) and \( (\alpha^*, \alpha_2^*) \) based on which one has an estimated impedance spectrum closer to \( Z_f \). Figure 3.6 shows the measured impedance profiles caused by one and two concurrent touches, respectively.
Figure 3.6: Measured impedance of a single touch ($\alpha^* = 0.68$) and two touches ($\alpha_1 = 0.66, \alpha_2 = 0.74$), shown in complex numbers with a real (resistance) and imaginary part (reactance).

Although I only demonstrate the detection and localization of two touches, my approach can be extended to more than two concurrent touches. I leave it for future research.

Calibration

The goal of the system calibration is to acquire all the necessary information (e.g., $Z_G$ and $Z_T$) needed to calculate the estimated impedance using the above equations. I calibrate $Z_G$ and $Z_T$ separately, because it allows the system to be aware of environment change based on the change in $Z_G$ alone, enabling new types of interactions (details later). Due to frequency sweeping, I use $\tilde{Z}_G$ to denote the impedance spectrum of the ground coupling effects measured under all the N frequencies (or background profile). I use $\tilde{Z}_{T1}$ and $\tilde{Z}_{T2}$, to denote the spectrum for the finger and contact impedance for one and two touches respectively (or touch profile). Finally, $Z_0$ is the impedance of the thread without touch (or thread impedance), which can be estimated using resistance, measurable using a multimeter, by ignoring the small effect of capacitance and inductance.

**Calibrating Environmental Noises:** As suggested by [74], the effect of ground coupling can be approximated using a distributed capacitance, which can be calculated using the following formula:

$$\tilde{Z}_G = \frac{k(1-k)\tilde{Z}_0^2}{\delta\tilde{Z}_C}$$
where \( k \) is a constant between 0 and 1 and \( \delta \tilde{Z}_C \) is the increase in the measured impedance caused by the coupling effect. As suggested in [74], I used 0.21 for \( k \) in my calculation. Although this approach overlooks the other parasitic components that may introduce noises, such as the impedance of electrodes at the two ends, I found these noises have a negligible impact on system performance in my later experiments.

Note that calibrating the background noises does not need to be carried out manually by a user. Instead, my system is designed to be re-calibrated automatically and periodically (e.g., every 5 minutes). By cross comparing every new background profile with the ones stored in the database, the system can identify if the user is in an unknown environment. This allows the system to notify the user for recalibration if the user is in a new environment.

**Calibrating Finger and Contact Impedance:** This step of calibration requires a user to touch a predefined location on the thread using the index finger. This is to allow the system to calibrate \( \tilde{Z}_S \). Again, \( \tilde{Z}_S \) approximates \( Z_G \) and \( Z_T \) connected in parallel, which can be calculated using the following equation, with a pre-determined \( \alpha \). In my case, \( \alpha = 0.5 \) as I asked the user to touch the middle of the thread.

\[
\tilde{Z}_S = \alpha (1 - \alpha) \frac{\tilde{Z}_0^2}{\tilde{Z}_f - \tilde{Z}_0}
\]

Since \( \tilde{Z}_S \) approximates \( Z_G \) and \( Z_T \) connected in parallel, when both \( \tilde{Z}_S \) and \( \tilde{Z}_G \) are known, a single-touch profile \( \tilde{Z}_{T1} \) can be calculated using the equation

\[
\tilde{Z}_{T1} = \frac{\tilde{Z}_G \tilde{Z}_S}{\tilde{Z}_G - \tilde{Z}_S}
\]

Like the background profile, the user profile can be stored in a database for repeated usages. The touch profile for two fingers can be retrieved in the same manner but with two predefined location coefficients. I show the equations needed for computing the profile for the two touches in the appendix. Finally, I show in Figure 3.7 the touch profiles of a user.
touching the thread using one vs. two fingers.

Figure 3.7: Touch profiles of a user touching the thread using one ($\alpha = 0.68$) vs two fingers ($\alpha_1 = 0.66, \alpha_2 = 0.74$).

**Calibrating Sensor Impedance:** An important requirement of my system is the uniformity of the resistance across the length of the sensor. However, this requirement may not always be satisfied (e.g., due to issues in the manufacturing process). As such, extra calibration can be used to obtain a more accurate impedance curve. The calibration can be performed by a user touching extra pre-determined locations (e.g., three evenly spaced points across the sensor). Once the extra calibration data is collected, a precise mapping can be established between the calculated location coefficient $\alpha^* s$ and the ground truth positions. The curve segments between the calibration points can be interpolated linearly.

### 3.3.5 Implementation

To evaluate my approach, I developed a proof-of-concept prototype using a 3D printer and off-the-shelf hardware and software. This section presents my implementation details.

**Thread Prototype**

Like in the previous work [53], the resistivity of the conductive thread cannot be too high or too low. This is because if the resistivity is too high, the electric field will be too weak to sense the signal. However, if the resistivity is too low, the change in the difference in the impedance caused by the finger touching the thread can be too small to be detected. I
found resistances between 500 k and 1 M work the best for my impedance analyzer chip (AD5933) from Analog Devices [75]. This requirement unfortunately makes most, if not all, the commercially available conductive threads unqualified for my need. Therefore, I had to create my proof-of-concept prototype using a 3D printer. The thread was a thin line of conductive filament in 11.5 cm long with a 0.1 mm × 0.4 mm rectangle cross section (dimension limited by my 3D printer). The thread has a resistance of 432 $K\Omega$, coupled with a 470 $K\Omega$, resistor to extend the sensing range to approximately the length of the sensor. Ideally, the resistance of the thread and coupling resistor need to match or sensing range or accuracy may be affected. The thread was created using a 3D printer (Ultimaker 3) [76], with the carbon filament CDP12805 from Proto-pasta. The sensor is bendable and can preserve its resistance distribution relatively well when bent. The two ends of the thread are connected to my sensing board using copper wire (Figure 3.8).

![Figure 3.8: My thread sensor prototype, coupled with a 470$k\Omega$ resistor and connected to the AD5933 impedance sensing board.](image)

**Impedance Sensing Board**

The main components of my sensing board are an impedance analyzer chip (AD5933 from Analog Devices) [75], a 1MSPS analog to digital converter (ADC), and a digital signal processor (DSP), which runs discrete Fourier transform (DFT) and returns the real and imaginary parts of the measured impedance of the thread at a desired frequency. The AD5933 allows impedance measurement from $1k\Omega$ to $10M\Omega$ with a system error of around 0.5%.
The AC frequencies of my sensing board were from 10 kHz to 100 kHz with a 2 kHz interval. This leads to a total of 46 samples per cycle. When the sensor is working, a small AC voltage is applied between the two electrodes, and the current flow through the thread is measured using an auto balancing bridge circuit. The impedance between the two ends of the thread is then calculated as the ratio between the input voltage and the output current measured by the sensor. Ideally, the resistor should have the same resistance as the sensor, I were unable to find one on the market. Although this affected sensing accuracy, the implementation was sufficient to demonstrate usage scenarios.

3.3.6 Demo Applications

To showcase ThreadSense and its capabilities, I created four demo applications and highlight various usage scenarios. Each application demonstrates the use case of one or more of interactions enabled by my sensing approach.

Interactive Hair

I implemented my sensing technique on a braided headband to allow gestural input to be carried out as if the user is touching or scratching the hair (Figure 3.9). Although touch input on the hair has been explored in prior research [77], the state-of-the-art methods can only sense the occurrence of touch. With ThreadSense, the systems can also detect swipe gestures for an extended input vocabulary. The use case for this type of input is broad. In particular, I see its benefit of being less obtrusive in social settings. For example, in the situations where repeated interacting with a smartphone or watch (e.g., checking voicemail messages) can be considered inappropriate. Swiping the hairband is less interruptive to other people since the motion is ambiguous about whether the user is actively using technology or just touching the head. In my implementation, a single touch on the sensor swiping downward plays a voicemail message on the earbuds hangs up the phone call, and a swiping gesture changes the phone to mute mode again stops the message. Although there
is only one thread sensor in my current implementation, I foresee that the entire headband can be augmented with my sensing technique in the future with more research efforts. This will enable a much richer set of interactions via the hair.

Figure 3.9: A user performs a subtle swipe gesture on a braided head band to interact with a computing device. Subtle swipe gesture inputs on a head band.

**Interactive Embroidery**

The next application is an interactive embroidery that allows the user to perform input on a soft object covered or made by interactive fabrics. In my implementation, I manually sewed my thread sensor onto an owl embroidered cushion cover (Figure 3.10). Touching different locations on the thread triggers different actions on a smart IoT device. For example, touching somewhere near the first owl disables/enables the microphone of the Alexa. Tapping near the second owl using one finger turn on the music. Tapping near the same location using two fingers plays the news. Swiping near the same location navigates the menu. This way, the cushion becomes the user’s always-available remote controller at the couch and can be used when voice input is not desirable.
Interactive Wire Crafts

Another example of bringing rich interactivity to objects that are traditionally passive is through wire crafts. In my third application, I developed an interactive tulip bookmark. When in use, the bookmark has a new function that allows the user to perform continues touch input to control the brightness of the ambient light while reading. For example, the user can slide the finger to increase or decrease the brightness of a lamp (Figure 3.11).

Interactive Headphone Cable

Finally, I demonstrate that all the interactions can be integrated into a single device of a headphone cable to extend the input capability of the existing work that cannot sense touch locations [56]. With my implementation, the user can use a single tap to quickly pick up a call. The user can also record a call by tapping the cable using two fingers. Further, the user can use slide with a single finger to control the volume or two fingers to navigate the contact list. A unique feature of my system is that it can recognize the change in the user’s
environment through the auto-calibration process. When the system detects that the current $\tilde{Z}_{G}$ is significantly different from the ones in its database, it shines an red light using an LED to notify the user to recalibrate the finger and contact impedance ($Z_{T1}$ or $Z_{T2}$). Once the new environment is calibrated, the profile is stored in the system for later use, and the light turns green (Figure 3.12).

### 3.3.7 Evaluation

I conducted a system evaluation to measure the accuracy of my sensing approach in locating touch positions and distinguishing touches between one vs. two fingers.

**Participants**

Ten right-handed participants (5 female) between the ages of 20 and 25 participated in the study. The average width of participants’ index fingers is 14.47 mm; s.e. = 0.18).
Figure 3.12: a red light indicates that a user enters a new environment, so that a recalibration is needed. Right: two concurrent touches on the earphone cord to start recording. recalibration notification on an interactive headphone cable when a user enters a new environment. Right: two concurrent touch on the interactive cord to record a call.

**Apparatus**

As the sensor is extremely thin and bendable, keeping it straight is important for us to accurately record the touch position(s) as the ground truth. For the study only, I printed the sensor straight on a supporting structure made of non-conductive material. Additionally, I printed the sensor twice as long as its original size. The extra piece was used to replace the coupling resistor to ensure a precise measurement of my sensing accuracy. During the study, the sensor was placed on an empty wood desk in front of the participant, who performed the task in a seating position.

**Calibration**

Prior to the study, the system was calibrated for each participants using one and two fingers. In the one-finger condition, the calibration data was collected at five locations evenly apart from each other across the sensor (0mm, 23 mm, 46 mm, 69 mm, 92 mm from the left end) (Figure 3.13 top). The location distances are calculated using the end near the coupling resistor as the origin. The calibration data was used later to investigate whether and how
well the number of touch points involved in the calibration may improve sensing accuracy. In the two-finger condition, the calibration data was collected with the participant touching the sensor using the index and middle finger, approximately 8 mm apart from each other. Participants were asked to center a target position at 46 mm (from the left end) in the middle of the two fingers (Figure 3.13 bottom). Calibration for two fingers was only performed at a single location.

![Figure 3.13: Calibration locations for one (top) vs two fingers (bottom; 8 mm space between the fingers). The numbers indicate the order in which the calibration points were added for analysis on the effect of the number of the calibration points.](image)

**Data Collection**

*One finger.* For the touches using one finger, participants slide their index fingers against the narrow edge of the sensor (0.1 mm) from 0 mm to 92 mm two times, with a sliding distance of 92 mm each. The start and end positions were chosen to avoid them mistakenly touching outside the sensor’s sensing region. Participant’s right index finger was held in a ring mounted on a slider placed in parallel to the sensor. This allowed them to precisely control the position of the finger on the sensor. Participants stopped every 2 mm, and the experimenter recorded the ground truth, measured using a ruler (Figure 3.14). A computer recorded the predicted location.

*Two fingers.* For the touches using two fingers, participants repeated the same task used in calibration but at three locations: 23 mm, 46 mm, and 69 mm. At each location, they were asked to center the target location in the middle of the index and middle finger, with
the fingers set apart in a distance of 0 mm, 4 mm, 8 mm, 12 mm, or 16 mm from each other (Figure 3.13). Zero distance means the two fingers were touching each other. Each trial was repeated twice. In total, I collected 920 and 300 samples for the touch events using one and two fingers respectively.

**Single Touch Accuracy**

I used average error distance ($ED_{avg}$) to measure the tracking accuracy of my approach. The $ED_{avg}$ per participant is defined as  \[
\frac{1}{n} \sum_{i=1}^{n} (|\tilde{y}_i - y_i|),
\]  where $\tilde{y}_i$ is the predicted location, and $y_i$ is the ground truth, and $n$ is the total number of trials per location per participant (e.g., 46 locations × 2 repetitions). I then average the $ED_{avg}$ across all participants as the final accuracy metric.

**Sensing Accuracy with Mid-Point Calibration:** I first looked at the sensing accuracy achieved using a single calibration point in the middle of the sensing region. Figure 3.15 plots the $ED_{avg}$ for the 46 tested locations (0 mm to 92 mm, stopped every 2 mm) and the region covered by the stand error. I observe that $ED_{avg}$ was 6.13 mm (s.e. = 0.26 mm) across all $ED_{avg}$. Additionally, the average error distance is higher in the first quarter of the sensing region ($ED_{avg} = 5.21$ mm; s.e. = 0.26) than the remaining part ($ED_{avg}$). Since half
of the thread is used as a coupling resistor, this first quarter of the sensing region is close to the half point that separates the thread into two equilibrium impedance ($\alpha = 0.5$). As shown in Figure 3.4, impedance change caused by the touch location is much less significant when the finger touches the thread center (i.e., near the peak of the curve in Figure 3.4). It results into slightly coarser-grained granularity in differentiating touch locations and thus larger location errors.

**Effect of Calibration Points:** I further compared the sensing accuracy while varying the number of touch points used for calibration. More specifically, with the two-point calibration, the calibration points were picked at the middle and start position; the three-point calibration was similar but with one additional point picked at the right end of the sensor; the four-point calibration was based on the three-point version with one additional point picked between point 1 and 2. (Figure 3.16).

As shown in Figure 3.16, sensing accuracy improved with the increase of the number
of calibration points along with the sensor. Note that error may take place on either side of the touch location. Therefore, a touch sensed within the distance of $ED_{avg}$ to both side of a touch button should be considered a successful hit on the target. Similarly, the distance between two touch buttons should be at least twice as big as $ED_{avg}$. I were able to reduce the $ED_{avg}$ to 4.16 mm with five calibration points. With this level of sensing accuracy, continuous input through sliding is possible but a filter needs to be used to reduce noise. The improvement brought by the calibration points near the fourth quarter (point 3 5) is smaller than that brought by the calibration points near the second quarter (point 2 4). I expect that point 3 5 is not needed if the resistance of the thread is evenly distributed. The result suggests an improved calibration strategy with more points near the coupling resistor is preferred.

I further examine the location accuracy across participants. Figure 3.17 plots the average location error and the standard error for each participant. I observe that the $ED_{avg}$ for most participants are around 5 to 8 mm. Participant 4 has the largest error because their finger width (17.69 mm) is much larger than the average width (14.47 mm), which influenced the ground truth reading. Overall, my results suggest that touch controllers designed for ThreadSense are likely generalizable among different users.

![Figure 3.17: The $ED_{avg}$ using the mid-point calibration for the ten participants.](image)

Two-Finger Touch Accuracy

My result shows that $ED_{avg}$ is 8.55 mm (s.e. = 2.17 mm) across the three locations and 15 touch points (Figure 3.18). It is 2.42 mm higher than the error of one-finger touch achieved using the mid-point calibration. Error is again higher near the first quarter of the sensing region with the $ED_{avg}$ of 10.29 mm (s.e. = 1.95 mm), 7.95 mm (s.e. = 2.36 mm), and 7.41 mm (s.e. = 0.39 mm) for the tested locations at 23 mm, 46 mm, and 69 mm. As shown the Figure 3.18, the error also increased with the increase of the distance between the finger,
except at location 46mm where the calibration point (8mm spacing) has the lowest $ED_{avg}$. I suspect that it is because the touching area of the finger changes with the opening of fingers, which lead to small deviation in the $Z_{T2}$ from the calibrated one.

![Figure 3.18: The $ED_{avg}$ of two concurrent touches by location and spacing.]

**Distinguishing One vs. Two Fingers**

Besides the tracking accuracy, I also investigated how well my system can distinguish one versus two concurrent touches. I used the same data in the accuracy evaluation but fitting the data using the models for both one and two fingers. The recognition was carried out by comparing the similarity between the estimated and measured impedance spectrum. Out of 920 samples collected using one finger, three were recognized as two fingers (error rate: 0.32%). The errors occurred at 69 mm, 85.1 mm, and 89.7 mm from left. Out of 300 samples collected using two fingers, three were recognized as one finger (error rate: 1%). One error occurred at 16.1 mm, 29.9 mm and the other two occurred at 52.9 mm, 85.1 mm. Most of the errors occurred near the end of the sensor, where the increase in impedance caused by touch is less observable.

### 3.3.8 Supplement Study

I conducted three supplementary studies to preliminarily evaluate how well the system can perform under different environmental noises, finger conditions, and distinguish between different environments.
Environmental Noise

The goal of this study was to measure the robustness of the system against different environmental noises. This study was carried out with a single participant (male, right-handed, 24 years old).

Data Collection: The study included four daily environments: (1) a running car, where the sensor was placed on the back seat; (2) outdoor, where the sensor was placed on a wood desk in an open space; (3) kitchen, where the sensor was placed on a dining table, surrounded by kitchen appliances, including a refrigerator and a microwave; and (4) workplace, where the sensor was placed on a wood desk, full of computers and electrical cables. In each environment, the participants taped the sensor using the index finger at five locations: 0 mm, 23 mm, 46 mm, 69 mm, 92 mm from the left. Each trial was repeated five times. In total, I collected 100 samples for data analysis. During the study, the sensor board was powered through a USB cable connected to a laptop.

I first measured the sensing accuracy with $\tilde{Z}_{T1}$ calibrated in each tested environment. I then measured the sensing accuracy with $\tilde{Z}_{T1}$ calibrated in an environment outside the tested ones – in a clean lab desk. This was to investigate whether and how well the system works in a new environment without recalibrating finger impedance.

Result: The data was analyzed using a repeated measures ANOVA with Environment and Calibration as independent valuables. My result showed that the $ED_{avg}$ across all the four environments was 3.96 mm with the system recalibrated separately for each environment. ANOVA showed no significant different difference between the four environments in the recalibration condition (p=0.93). In contrast, the $ED_{avg}$ increased to 25.63 mm without recalibration. ANOVA yielded a significant different difference between the condition with and without recalibration ($F_{1,76}=13.65, p<0.001$). The most noticeable difference between the two calibration conditions was found in the outdoor environment (Figure 3.19). This is mainly because of the large difference in the grounding effect caused by the earth (outdoor) and the floor of my building.
The $ED_{avg}$ from this participant was lower than the average sensing accuracy found in the main evaluation. This is fine as sensing accuracy varies among users. For this study, I were only interested in the difference between the tested conditions.

**Distinguishing Environments**

The goal of this evaluation was to measure how well the system can distinguish between different environments based on the difference in the current background profile ($\bar{Z}_G$) and the ones existing in the system database. I defined the similarity of two profiles using the Frechet distance between them. The smaller the distance is the more similar the two profiles are. My database contains 100 samples for each of the four environments tested in the previous study. To calculate the recognition threshold for each environment, I first calculated 100 distance scores by calculating the similarity between the each of the database profile and the average of all the 100 samples (average profile). The recognition threshold is thus the sum of the average and standard deviation of the distance scores. If the distance between the testing and the average profile is smaller than the recognition threshold, the environment is recognized. Otherwise, I tag it as a new environment. My testing data included five new samples from each of the four tested environments, collected in a 20-minute interval. My result showed a 100% recognition accuracy with no false positive.
Finger Conditions

Finally, I conducted another quick test to investigate if the sensing accuracy is affected by a sweaty finger. I repeated the test in the lab environment with the same participant performing the task with a dry or sweaty finger. To create a sweaty finger, I asked the participants to jog for thirty minutes. I collected 50 samples for data analysis. My result showed that $ED_{avg}$ was 2.99 mm in the dry finger condition but increased to 4.41 in the sweaty finger condition. This is expected as body and touch impedance change when the finger sweats. A more careful study is needed to better understand how system performance may vary with the change of body condition.

3.3.9 Discussion and Future Work

I discuss insights gained from this research, propose future research, and acknowledge the limitations of my work.

Spacing Resolution

The sensing resolution of my approach is affected by the operating frequency of the system. As suggested by Sato et al. [70], a wider operating frequency may likely introduce more observable changes in the signal spectrum in response to the user input. The spacing resolution can also be improved with an increased impedance per sensor length. This allows the signal caused by a small displacement of the finger to be more noticeable. Note that the sensor board used in my current implementation only supports an operating frequency of up to 100 kHz and maximum impedance measurement of $10M\Omega$. I expect that the sensing accuracy of my approach can be improved with better hardware.
Evaluation

I presented a technical evaluation that demonstrated promising results for sensing accuracy for my approach. The result should be considered as a high bar of what my sensing technique could achieve since the samples were collected under controlled conditions. Future evaluations should look at whether and how sensing performance may be affected if the sensor is bent. ThreadSense could also benefit from a formal user studies to understand how the device would be used by end users in the proposed scenarios. For example, a user study may help us find preferred gestures for these applications.

Sensor Fabrication

Although the sensor in my current implementation was created using a 3D printer, I believe that it can be manufactured using fibers. One possible approach is to coat a fiber thread using conductive ink. Another possibility would be utilizing materials like conductive polycarbonate to create the thread. The challenge of course is in the requirement for the conductive to be evenly distributed along the thread, which requires a precise manufacturing process to overcome. Possibility of User Recognition. The validation result shown in Figure 3.4 suggests that the impedance profiles differ across the users. This is primarily due to the difference in their body impedance. Such individual differences can potentially be useful for differentiating input from different people. The key challenge, however, lies in the mixture of the effects on measured impedance caused by the user’s body and the finger touching at different locations. I plan for future work to investigate methods that can separate these effects for user differentiation.

3.3.10 Conclusion

In this project, I discussed an approach to enabling 1D touch sensing on an interactive thread based on impedance sensing. My technique can locate up to two touches with a
spacing resolution unachievable by the existing methods. My system is also unique in that it employs a model-based approach, which locates touches based on a mathematical model describing the change in stripe impedance in relation to the touch location. By sweeping the frequency of injected current during impedance measurement, the system requires only a quick calibration without training. The simplicity of the calibration allows the system to quickly adapt to various environments and users. I foresee that the proposed approach can go beyond 1D touch sensing and serve as an important groundwork for future investigations in sensing techniques on extremely thin and soft objects.
3.4 Fabriccio

3.4.1 Introduction

With the rise of computational materials, the need to bring interactivity to soft and lightweight fabrics (e.g., garments, toys, and furniture) has grown significantly. This has led to advances in sensing techniques that enable explicit inputs to be carried out on computational textiles (i.e. interactive fabrics), such as touching [42, 43, 8] or deforming the fabric [44, 45].

However, a challenge with existing input modalities is that physical contact with fabric must occur during the interaction. Thus, opportunities are missed for users to utilize other methods, such as touchless (or mid-air) hand gestures, commonly seen on smartphones [78], smart watches [79, 80, 81], car infotainment systems [75], and smart IoT devices [82]. The touchless, mid-air gestures performed by a hand or fingers near a sensor, significantly extends the input vocabularies of interactive fabrics including those carrying special meaning that can’t be replaced by touch (e.g., waving the hand for a greeting). Touchless gestures are also useful in common scenarios where physical contact with a fabric is undesirable by a user (e.g., the hands are unclean when eating or exercising).

In this project, I bring near-field touchless gestural input to interactive fabric using doppler motion sensing. With my technique, soft objects augmented with a textile motion sensor can detect nearby finger gestures (e.g. in 10 cm distance [79]) to trigger a desired application. This enables new types of interactions in a variety of contexts. For example, a plush dog toy can make a greeting sound to respond to a child’s hand waving in front of it. When standing or walking, a user can perform micro finger gestures (e.g., sliding on the index finger using the thumb) with the hand hanging naturally alongside the body to discretely interact with a screen (Figure 3.20). This type of gesture is subtle, easy to perform, and now sensed through trousers, instead of needing heavy, leg mounted depth cameras which are used in current methods for such scenarios [83].
To demonstrate technical feasibility and application scenarios enabled by my technique, I developed a proof-of-concept prototype called Fabriccio (Figure 3.20). My prototype was developed using a pair of loop antennas (one as a transmitter and the other as receiver) made of a conductive thread that was sewn onto a fabric substrate. The antenna type, configuration, and operating frequency were carefully chosen to balance sensor sensitivity and the complexity and cost of the fabrication process, making it easy for the system to be adopted by the large maker community. My prototype detects 10 touchless gestures, involving hand and finger motions in different scales. It can also detect the finger tapping the sensor. Results from an evaluation with 10 participants revealed 92.8% cross-validation accuracy and 85.2% leave-one-session-out accuracy.

Figure 3.20: Sensing accuracy with and without calibration in different environments.

My contributions are: (1) a touchless gesture sensing technique for interactive fabrics that uses the Doppler effect; (2) a prototype demonstrating technical feasibility; (3) a study evaluating the accuracy my sensing technique; and (4) several applications to demonstrate the unique interactions enabled by my technique.

3.4.2 Related Work

My work intersects with previous research in three main areas: sensing technique for interactive fabrics, sensing techniques for touchless gestures, and textile antennas.
Sensing Input on Interactive Fabric

With current technologies, input techniques through interactive fabrics includes touch [46, 42, 43], and deformation [44, 45, 84], as well as object recognition [85, 86] using sensing techniques based on capacitance [46, 42, 45], resistance [43, 44, 84], and inductance [85].

Capacitive sensing is based on the effect of capacitive coupling and has been used in early explorations of sensing touch [42, 46] and pressure [45] on smart fabrics. For example, the Musical Jacket [46] from MIT features a capacitive touch keypad made of stainless-steel yarns embroidered on denim for a user to provide touch input. This technique was later used in other research prototypes [42, 54] but not until recently has it moved beyond research into commercial products. Project Jacquard [54] exemplifies a recent attempt to make the manufacturing process of capacitive sensing on fabrics scalable. With Project Jacquard, the electrodes of the sensor are created by weaving conductive yarn into a textile using a process compatible with the current industry standard.

Aside from capacitive sensing, techniques based on resistance are also common on smart fabrics. A textile resistive sensor has a three-layer structure involving a middle semi-conductive layer sandwiched between two conductor layers. eCushion [84] is an example of such an implementation. With resistive sensing, input is sensed based on the change detectible in the resistance of the fabric when the fabric is compressed. A wide variety of applications have been developed using resistive sensing. For example, in Rofouei et al.'s work [86] the authors used a textile pressure sensor for object recognition based on the pressure footprint of different objects (e.g., weight and shape). eCushion [84] was developed for sensing the sitting posture of a user on a chair. GestureSleeve [47] allowed users to use touch gestures on the forearm to interact with a computing device. proCover [87], an augmented prosthetic limb with pressure sensing capability uses a similar sensor. Recent advances in fabrication technique by Parzer et al. [43] allows the three-layer structure to be replaced by two thin threads.

Aside from resistance and capacitance, sensing techniques based on inductance have
also been explored on fabrics. For example, Jun et al.’s work allows a metallic object to be recognized when an object is in contact with fabric [85]. The same technique can be used for sensing touch but the input resolution in a 2D space is limited due to the coarse arrangement of the sensor coils. My work differs from existing research in that it brings touchless gestural input, commonly found in games, TV, vehicles, mobile, and, wearable applications to interactive fabrics using Doppler motion sensing.

**Sensing Touchless Gestural Input**

Sensing techniques for touchless gestural input can be divided into those based on vision [88, 89, 90], radio frequency [81, 91, 92, 93, 80, 94, 95] and pyroelectric infrared [79] and acoustics [96, 97, 98]. Cameras (both 2D and 3D) are also often used in a wide variety of applications. Examples include the work from Song el al. [89], which enables the sense of gestural input in a 3D space using a 2D camera, and the work from Wang et al. [90], which tracks 6DOF bimanual hand input using a depth camera. In addition to the vision-based approaches, techniques using radio frequency have also shown promise in sensing touchless hand (e.g. flicking, sliding, or hovering) [93, 91, 80] and finger gestures (e.g. pinching the thumb and index finger, pinching the thumb and pinky, sliding the thumb along the index finger, or rubbing the thumb and index finger) [79, 81, 80, 99]. Examples in this line of work include Mudra [94], which detects finger gestures using home Wi-Fi signals, and Soli [81, 80], which detects hand and finger gestures using 60 GHz radar signals. My sensing technique is based on the Doppler effect, which has been shown effective in sensing hand motion as input for mobile devices [95]. Along with these methods, techniques based on pyroelectric infrared [79] and acoustics [96, 97, 98] are pushing the boundary of touchless gesture sensing. However, the challenge in adopting these methods on soft and thin fabrics, is that existing methods are developed on traditional devices with a rigid body (e.g., the sensor is printed on a PCB) and thus do not immediately work on a fabric.
Textile Antennas

Textile antennas made of conductive threads are an emerging technology in electrical engineering with applications primarily targeting wireless communication [100, 101, 102, 103, 104, 105], health monitoring [106, 104, 107], tag identification [108, 109], and energy harvesting [110, 111]. For example, Roundjane et al.’s work [104] describes a spiral-shaped textile antenna stitched on a T-shirt for transmitting Bluetooth signals at a frequency of 2.4G Hz. Placed on the chest, a textile antenna can be used to sense the wearer’s breathing rates using Bluetooth and received signal strength indicator (RSSI). Shao et al. [109] proposed a textile RFID tag for object recognition. Loss et al. [111] developed a monopole antenna to harvest electromagnetic energy from the GSM and DCS signals in the environment. My work is novel in that I are the first to investigate how a textile antenna can be designed and developed for touchless gestural input. I identified the challenges unique to this problem and demonstrate a promising solution for a new set of smart fabric applications.

3.4.3 Sensing Principle and Background

Doppler motion sensors are known for being cost-effective sensors for in-air gestures [112, 91, ？]. Its sensing principle is based on the Doppler effect, described as the shift in the frequency of a wave caused by the motion of an object (e.g., hand) in relation to the wave source. Most Doppler motion sensors have a transmitter and receiver, with each connecting to an antenna via transmission lines. The antennas are often placed next to each other at a certain distance. When operating, the transmitter supplies an electric current to the transmitting antenna, which radiates energy from the current as electromagnetic waves through the air. When there is a moving object near the sensor, the receiving antenna intercepts some of the power of the electromagnetic waves reflected by the object and produces an electric current to the receiver. The receiver then mixes the reflected signal with a local signal of the baseband frequency, resulting in the intermediate frequency (IF) signal. The shift in the frequency of the reflected waves can thus be observed in the IF
signals. – in my case, for interaction purposes.

The resolution of the frequency shift of the Doppler motion sensor is related to the operating frequency. The higher the operating frequency is, the more observable the shift in the reflected frequency will be. The sensitivity of the sensor is dependent on the signal-to-noise ratio (SNR), and often related to the strength of the received signal. In designing a textile Doppler motion sensor under a certain operating power, the antenna type, the distance between the antennas, and how the antennas are connected to the transmitter or receiver may significantly affect the sensitivity of the sensor. My work strikes a balance between the sensitivity and fabrication cost and complexity.

3.4.4 Sensor Design

In this section, I present the design of my textile Doppler motion sensor based on four parameters: sensor operating frequency, antenna type, transmitting/receiving antenna configuration, and impedance matching.

Operating Frequency

For my implementation, I considered an operating frequency of 1 GHz and above for the sake of sensing resolution. In this range, three bandwidths are common in commercial Doppler sensors that comply with the FCC regulations: X band (10.525 GHz), K band (24 G to 26 GHz), and V band (60 GHz to 67 GHz). The high frequency antennas are in general good in resolution but extremely challenging to develop on fabric because of the level of precision needed in the fabrication process. For example, the diameter of a loop antenna running at 60 GHz must be made precisely at 1.59 mm (circumference of a loop antenna equals to wavelength). A small error of even +0.5 mm in diameter will shift the antenna’s operating frequency dramatically to 45.7 GHz [113]. To increase the accessibility of this technique to other researchers and makers, I instead use X band (10.525 GHz) in my exploration because the X band antennas are relatively larger in size and can be made in a
level of precision that is achievable using a low-cost home embroidery sewing machine. I restricted my system to work at 3.3v, as I considered the capacity of the batteries in toys and wearable applications.

**Antenna Design Options**

For touchless gesture sensing, a desirable transmitting antenna design is one that can radiate a strong electromagnetic field above it. The challenge is that no existing knowledge provides an insight into the tradeoffs of the possible design options under my application requirements (e.g., X band, 3.3v, and near-field sensing). I thus conducted a simulation test, an approach commonly used in the design of textile antennas [114, 115, 116, 117, 118]. I considered four common antenna types found in the literature [115, 103, 116, 117, 118], including dipole, loop, patch, and slot antenna. I created the candidate antennas in COMSOL [118] by following the size requirement for them to operate at 10.525 GHz (details later). Like prior work [119], I simulate the antenna using copper fabric (conductivity = 0.05Ω/square). I am aware that the simulation may not replicate the antenna behavior on a real fabric, but the comparison using the estimation of the electric field served well for my decision making. Table 3.1 shows the electric field for each candidate in a 10 cm × 10 cm space.

**Dipole antenna:** The dipole antenna consists of two traces of equal length (1/2 wavelength), oriented end-to-end on a substrate. The structure of the dipole antenna is simple and can be made using conductive threads, embroidered onto a fabric [103] (Table 3.1a). However, the electric field of the dipole antenna appears to be the weakest amongst all four candidates.

**Loop antenna:** The loop antenna features a simple structure with a circular trace, where the circumference equals to the wavelength of the operating frequency (28.5 mm in my case). My simulation results suggest that the electric field of the loop antenna is stronger than dipole antenna. The loop antenna can also be fabricated using low-cost home embroidery sewing machine [104].
Table 3.1: The antennas and the electric field radiated by them in a 10 cm × 10 cm space.

**Patch antenna:** The patch antenna consists of three layers, with the top and bottom layer made of a conductive plate, serving as the radiating and grounding plane respectively. The middle is an insulating dielectric layer. The size of the patch antenna at 10.525GHz is 6.04 mm × 8.59 mm [113]. The complexity of the three-layer structure, as well as the harsh material and thickness restriction for the insulation layer, makes it difficult to develop on fabric [104]. Table 3.1g shows that the strength of the electric field above the patch antenna is similar to that of the loop antenna.

**Slot antenna:** The slot antenna is made of a conductive surface with a slot cut out. The size of the slot antenna at 10.525GHz is 14.25 mm × 1.34 mm [113]. Table 3.1h suggests that the electric field radiated by the slot antenna is strongest of all the four candidates. The transmission lines of the slot antenna need to go through the center of the slot, thus an isolation layer needs to be placed between the transmission lines and the antenna.

**Summary:** Considering the balance between fabrication complexity and strength of the electric field, I chose the loop antenna in my implementation for both the transmitter and receiver. Note that the loop antenna is bidirectional, which means gestures can be sensed on both sides of the sensor, enabling new types of interactions. However, if only one direction is needed, an insulation layer can be used (more details later).
Transmitting and Receiving Antenna Configuration

The next step is to understand the impact of antenna configuration (or distance) on signal strength. If the antennas are placed too close to each other, the electric field of the transmitter may be interfered with by the receiving antenna, thus weakening the signal. Moving the antennas away from each other can eventually solve this problem but may also weaken the signal. I thus conducted a second simulation test.

**Simulation Setup:** I studied 20 configurations with the antenna distance ranging from 1 mm to 41 mm, with a 2 mm step size. The distance was calculated using the closest points on the two loops. 1 mm was chosen as the closest distance because it is the closest that two threads can be stitched without touching each other using my embroidery machine. The other parameters, such as material type, remained the same as the first simulation study. Since I were only interested in comparing the strength of the reflected signal, my simulation did not need a moving hand to create the reflection. Instead, I used a virtual circular copper plate above the antennas to create the reflection. I adjust the size of the plate to reflect the difference in the size of the hand and finger. Considering the average width of a hand and finger [120], I used a plate of 10 cm and 1.8 cm wide for the hand and finger scenario respectively. My data was collected with the copper plate placed at 5 cm or 10 cm above the sensor. The larger plate was positioned to cover both antennas, but the smaller plate was not big enough to cover both. Thus, I included three horizontal locations for the small plate, with one at the center of transmitting antenna, another at the center of the receiving antenna, and the last in the middle of the two antennas. In total, I sampled 20 distances × 2 heights × (1 location for the large plate + 3 locations for the smaller plate) = 160 data points for the test.

In COMSOL [118], the signal was represented using a complex number. Therefore, the strength of the reflected signal was calculated as the magnitude of the difference between the signal received with and without the copper plate. The collected data was then normalized across antenna distances and means (signal strength score) were calculated across the
tested conditions for each antenna distance.

**Results:** I show the study result in Figure 3.21. Additionally, I show the corresponding electrical field of the sensor with a step size of 4 mm. The peak of the signal occurred when the distance between the antennas was around 15 mm to 21 mm. The strength of the signal declines with the antenna distance exceeding 21 mm and beyond. On the other hand, the strength of the signal declines with a steeper slope before the peak, with the antenna distance becoming closer. This is due to the interference in the electric field from the receiving antenna. As shown in the bottom of Figure 3.21, the interference is clearly visible when the antenna distance is below 13 mm. Considering the balance between signal strength and sensor size, I chose 15 mm distance in my implementation.

![Figure 3.21](image)

Figure 3.21: Top: the signal strength score of the reflected signal on the receiver. Bottom: the corresponding electric field at different antenna distances with a step size of 4 mm, indicating the interference from the receiving antenna.

In summary, my final design includes two loop antennas placed 15 mm away from each other (Figure 3.22). The antennas have a diameter of 9 mm. A feed-point distance of 1 mm was used for the parallel transmission lines.

**Impedance Matching and Transmission Line Routing**

Energy loss may occur if there is a mismatch in the impedance of the antenna, its transmission line, and the transceiver [113]. Unfortunately, impedance match can hardly be
guaranteed on a textile antenna and transmission lines [121]. I mitigated the issue by restricting the length of the transmission lines to a multiple of half of the wavelength of the operating frequency (e.g., k × 14.25 mm) [113]. In my case, this method has an acceptable energy loss of around 11%, calculated using the formula of reflected power [113] with an antenna impedance of 100Ω [122], and source impedance of the transceiver circuitry of 50 Ω (by most design). The challenge of this approach, however, is that sensor applications may require the antennas to be at any location on a substrate. When turns are made along the way toward the target location, the curve line is longer in the outer track of a turn than that in the inner track. As such, one transmission line will fail to satisfy the length requirement.

I solved this problem by including a semicircle and two quadrants in the track of the transmission line (Figure 3.22). The semicircle and quadrants should have the same radius, but their direction should be reversed to ensure that the total length of the inner and outer transmission lines is equal after turns occur. Assuming that in a coordinate system, where the origin is the terminal of the RF transceiver, given the location of the feed point (x, y), the length of a transmission line connecting the origin and the feed-point can be described using the following equation:

\[
\frac{(k\lambda)}{2} = x + y + (\pi - 2)(2r - g) + 2l
\]

where \(k\) is the factor, \(\lambda\) is the wavelength (28.5 mm), \(r\) is the radius (\(r < y\); 3 mm in my case), \(g\) is the distance between the two parallel transmission lines (1 mm), and \(l\) is the length of the line segment connecting the semicircle and a quadrant, which is the only unknown valuable in the equation. As \(l\) is the function of \(k\), it can be calculated for any given \(k\) specified by a user using a variation of the above Equation:

\[
l = \frac{(x + y + (\pi - 2)(2r - g) - (k\lambda)/2)}{2}
\]
The same approach can be used for two antennas. Figure 3.22 illustrates my implementation.

Figure 3.22: My design for the transmission line includes a semicircle, two quadrants, and several straight-line segments. Conductive thread options.

### 3.4.5 Implementation

In this section, I discuss implementation details of the thread choice, hardware and software.

#### Conductive Thread Options

Another challenge of developing textile antenna is the choice of conductive threads. For example, the conductivity of the threads must be high, or energy loss may cause the reduction of the sensitivity. Additionally, the threads also need to be thin. Otherwise, the precision of the antenna traces cannot be guaranteed using a standard home embroidery sewing machine, further weakening the sensitivity. The conductive threads used in prior research [85] satisfy my needs (details in Table 3.2). Like in the previous work [104], I estimate the performance of these threads using a simulation. I repeated my second test by simulating the antennas made by the candidate threads using the same thread conductivity. My results showed no noticeable difference between the threads (Table 3.2). Considering that the higher conductivity the thread is, the less energy loss will occur in transmission, I used the LIBERATOR 40 in my implementation.
<table>
<thead>
<tr>
<th>Name</th>
<th>Manuf.</th>
<th>Material</th>
<th>Thickness (mm)</th>
<th>Conductivity (Ω per m)</th>
<th>Signal Strength Score</th>
<th>Electric field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stainless thin thread</td>
<td>Adafruit</td>
<td>316L Stainless steel fiber</td>
<td>0.20</td>
<td>51.18</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>Conductive thread</td>
<td>Sparkfun</td>
<td>12UM Stainless steel fiber</td>
<td>0.12</td>
<td>27.00</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>LIBERATOR 40</td>
<td>Sparkfun</td>
<td>316L Stainless steel fiber</td>
<td>0.35</td>
<td>91.84</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Syscom</td>
<td>Silver coated polymer</td>
<td>0.18</td>
<td>3.28</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Different types of conductive yarns tested in my simulation with the corresponding electric field shown in a 10 cm × 10 cm space. The signal strength score was calculated using the same way as in the second simulation test.

**Hardware**

**Fabricating Textile Antennas:** Once the antenna design and choice for the conductive thread is finalized, the textile antenna can be stitched using a standard home embroidery sewing machine (e.g. Brother SE600) on a fabric substrate (e.g. polyester in my case). I used stitching in my implementation as the antenna traces created using stitching can be mechanically stable and durable [85]. Note that the standard stitching process on an embroidery sewing machine pushes the conductive threads through the substrate, which may cause a short circuit if an insulation layer is used for unidirectional sensing. I adopted a method discussed in Dunneet. al.’s work [123] to overcome this challenge, where I carefully tuned the tension of the top thread (e.g. non-conductive thread) to ensure that the conductive thread only floats on the surface of the substrate without penetrating it (Figure 3.23).

**Customized Sensing Board:** My customized sensing board is composed of a Doppler sensor board and a data collection board (Figure 3.24). The Doppler sensor board was modified from the HB100 Doppler Radar Motion Detector by fully removing the patch
antennas and the transmission line traces from the PCB, leaving only the RF transceiver part. Ideally, the distance between the vias for the transmitter (or receiver) and ground should be the same as the antenna’s feed-point distance (e.g., 1 mm), however this is not the case with my off-the-shelf device (e.g., 1.5 mm). This may affect sensitivity but can be fixed in the future with a fully customized board. The Doppler sensor board operates at a frequency of 10.525 GHz.

My data collection board is composed of a differential amplifier circuit with a gain of 30 dB, which amplifies the data received from the sensor board to the range of 0v to 3.3v, a micro-controller with an built-in 10bit ADC (ATMEGA32U4), and a Bluetooth low energy module (nRF51822) for data transmission (Figure 3.24). The data collection board was mounted on the sensor board with the entire system operating at 3.3v with a sampling rate of around 1000 Hz. All the sensor data was sent to a laptop for data processing. In total, the entire system consumes 158mW of power including those consumed by the micro-controller and Bluetooth radio (45 mW). With a 400 mAh lithium-polymer battery, the system can work for approximately 2.5 hours. The cost of the sensing board is less than 30 dollars.
Figure 3.24: The customized sensing board consisting of a modified Doppler sensor board and a data collection board with a 400 mAh battery.

**Wire Connection:** Connecting the conductive threads to rigid electronics is an open problem in research that is yet to be solved [85, 43, 42]. Prior work has suggested methods, including soldering, using snap buttons, sewing, conductive epoxy, and crimping [85, 124, 43, 42]. Most of them except conductive epoxy, however, do not work in my case for varying reasons. For example, as suggested by [85], solder heat can make the connecting tip of a thread fragile, causing unstable connections between the transmission line and the vias. In my implementation, I used a low temperature solder paste. I first push the tip of the thread into the via and then soldered it with the solder paste using a heat gun at a temperature of 140°C. I also adhered the sensing board on the fabric to avoid parts moving at the connection points. My initial test suggested that this type of connection was stable, and durable in my experiments.

**Software**

**Signal Processing Featurization:** Sensor data was filtered with a cut-off frequency at 100Hz to remove the background noise induced by the powerline. The features for machine learning were extracted in both frequency and time domains. For the features in the frequency domain, I first computed a frequency spectrum using a 90% overlapping, 240 window-sized FFT, which is then used to compute the max, mean, min, and standard deviation for each frequency band (20 × 4 values), resulting in 80 features for the classifier. In
addition, I used a feature extraction toolbox (tsfresh [2]) to compute numbers of features in both frequency and time domain (e.g. continuous wavelet transform, the quantiles, binned entropy and etc.) In total, 480 features were fed into the machine learning model.

**Machine Learning:** To classify touchless gestures, I used the Random Forest from Scikit-learn with a forest size of 100 and the maximum depth of 30. I chose Random Forest because I found it more accurate in my initial tests than alternatives (e.g. SVM and Neural networks). The value of the parameters (e.g. forest size and depth) was chosen to balance the sensing accuracy and model complexity. I ran the classifier on a Microsoft Surface laptop.

### 3.4.6 Applications and Scenarios

I created several demo applications to elucidate Fabrriccio capabilities and highlight many of its usage scenarios in everyday furniture, clothing, and soft objects.

**Interactive Furniture**

The first application I implemented is an integrated media controller for a sofa, where a user controls the media playing on a TV, with gestures performed above an armrest. A swipe gesture can navigate the program, while a push gesture can pause or play media currently playing on the TV. In the scenarios where a user does not want to touch the sofa because their hands are unclean (e.g., when eating), the touchless hand gestures are useful additions to touch input on fabric (Figure 3.25a).

**Interactive Clothing**

My second scenario involves turning everyday clothing into interactive wearables. For example, I augmented the logo of a sports shirt with Fabriccio to allow a user engaged in a fitness activity to receive different types of audio information using gestures. For example, a user performing a check-mark gesture above a logo can be used for checking the percent-
Interactive furniture. (a) A touchless remote control on the armrest of a sofa. (b) A touchless environmental control on the tablecloth.

age of their fitness goal, and similarly, a thumb circle gesture can be used for listening to their fitness schedule through the headphone, (Figure 3.26). In another example, I instrumented a pair of pants with Fabriccio on the side. This allows a user to perform subtle arms-down gestures alongside the body to interact with a smartphone (Figure 3.20).

Interactive Soft Things

Finally, I demonstrate how Fabriccio can be useful in scenarios involving IoT-like devices by modifying everyday fabric-based objects. For some people, the backpack is a common part of life, and are used for carrying common objects, like smartphones. However, in some weather conditions (e.g. snow or rain) it is not ideal to take a smartphone out for simple tasks like answering a call or responding to notifications. I modified the two straps of a backpack by sewing and covering the sensors on each strap to allow for dual gesture input (Figure 3.27). For example, a circular gesture on the right strap allows the wearer to perform a circular gesture to listen their last text message when they are listening to music, while swiping near the left strap allows them to play and stop the music.

Interaction is an important part of children’s toys. I embedded Fabrricio into the head of a plush dog to enable simple interactive games for children. Waving the hand near the dog triggers a greeting sound. Touching its head plays a prompting sound (Figure 3.28).
3.4.7 Evaluation

The goal of this study is to validate Fabriccio’s gesture recognition accuracy, as well as its robustness against individual variance and among different users.

Gesture Sets

To ensure that my exploration covered a wide variety of different type of touchless gestures, I surveyed existing work and chose 10 hand and finger motion gestures (Figure 3.29). The gestures selected in my work vary not only in motion trajectory but also in the size of the motion, as many of the finger gestures, such as check mark and rectangle mark, are classified as micro gestures in the literature. I also include a touch gesture, which requires a user to tap the sensor.
Participants

Ten right-handed participants (average age: 21.6, 6 females) were recruited to participate in this study.

Data Collection

Each participant was instructed to sit in front of my textile sensor placed on a desk. Before a session started, participants were given several minutes to learn the 11 gestures. After the short training session, each participant performed the gestures toward the textile sensor roughly in the distance of 5 cm to 10 cm using their right hand. The order of gestures was randomly assigned. The start and end of each gesture was indicated by clicking a computer mouse using their left hand. Each gesture was repeated 10 times in each session, which took about 30 minutes to complete. A 10-minute break was given between sessions, where participants were asked to leave the desk and walk around the lab. Data collection finished after three sessions. In total, I collected 3300 samples (10 participants × 11 gestures × 10 repetitions × 3 sessions) for analysis.
Results

To demonstrate the accuracy of my system, I present my result using within-user accuracy, cross-section accuracy and cross-user accuracy. Also, I computed the SNR for each gesture to demonstrate the sensitivity of my textile sensor.

**Within-User Accuracy:** Within-user accuracy is the measurement of the prediction accuracy where the training and testing data are from the same user. For each participant, I conducted a two-fold cross validation, where half of the data was used for training and the remaining data for testing. The overall within-user accuracy was calculated by averaging the results from all the participants. The result showed an accuracy of 92.8% (SD = 3.6%). Figure 3.30 left shows the confusion matrix. The major source of error was the confusion between the finger gestures with a similar motion. For example, Click, Thumb slide and Thumb Check accounted for the most misclassifications, as they all have two sharp turns in the motion trajectory.

**Cross-User Accuracy:** Cross-section accuracy measured how stable the system was against the data collected from a different session. I calculated the leave-one-session-out accuracy for each participant by training the model using the data from the first two sessions
Figure 3.29: My gesture set. Click, rub, thumb slide, swipe, pull, push, and circle were chosen from Soli [81, 80]; Thumb circle, check, and rectangle were chosen from Pyro [79].

and testing it using the last session. The overall across-section accuracy was the average of the accuracy from all participants. The results yielded an accuracy of 85.2% (SD = 10.4%).

Figure 3.30 right shows the confusion matrix. Again, I found Click (82%), Thumb slide (76%) and Thumb Check (65.0%) contributed to the most errors. In addition, some finger gestures (e.g. Thumb Rectangle) began to cause confusions with others (e.g. Rub and Thumb Check). A potential reason is that the position and orientation in which the gestures were performed in relation to the sensor changed more significantly between sessions. I expect this issue can be mitigated with more training samples.

**Cross-User Accuracy:** Across-user accuracy measured whether an existing model works across different users. For the accuracy, I calculated the leave-one-subject-out cross-validation accuracy by using the data from nine participants for training and the remaining one for testing. The overall accuracy is the average of the ten combinations of training and test data. The results yielded an accuracy of 65.5% (SD = 6.6%), indicating that the different users performed gestures differently. For example, some participants performed Click by moving both the thumb and index finger, while others only moved their index finger with
the thumb staying in a relatively fixed position. Figure 3.31 shows the confusion matrix of all gestures. The most confusing gestures are Click (33.3%), Thumb Check (29.3%) and Thumb Slide (36.0%), followed by Thumb Rectangle (56.3%) and Circle (68.0%). I then removed them one by one and calculated the accuracies using the remaining data. The result yielded a higher accuracy of 75.3% (SD = 4.9%) without Click and Thumb Rectangle, and 87.6% (SD = 5.0%) without Click, Thumb Check, Thumb Rectangle and circle. This is encouraging, as the results showed that the differences in how the gestures (7 in my case) were performed across different people can be tolerated.

**Sensor Sensitivity:** To validate the sensitivity of my textile sensor, I computed the SNR for the samples classified successfully in the within-user validation. As shown in Figure 3.32, the SNR for all the gestures were above 3dB, indicating that a minimum SNR of 3dB and above is needed for the sensor to reliably capture the gestures performed at a distance around 5cm to 10 cm.
3.4.8 Effect of Covering Fabric

In applications, where a fabric cover may be used on top of the sensor for design or aesthetic reasons, attenuation may occur due to occlusion and the sensor signal may become too weak to be used for reliable gesture recognition.

To explore this issue, I collected through-fabric sensor data using 17 fabrics made of acrylic, cotton, jute, linen, nylon, polyester, PVC leather, PU leather, faux fur, rayon, T-spun polyester, spun polyester, polypropylene, lyocell, olefin, modal rayon, and metallic yarns. I selected fabrics that are commonly found on garments, furniture, toys, and upholstery. I purposefully included those with varying thickness (0.17mm to 1.62mm) and materials (e.g. some have metallic threads). For each fabric, I measured the signal strength of my sensor in response to an object moving in a consistent pattern in front of it. I shielded the sensor on the back using a copper plate to avoid noise coming from the back. The consistency of the object movement across the tested fabrics is important to make an accurate comparison of the sensor signal. My method for controlling the consistency was to use a motorized aluminum plate (20 cm × 20 cm) moving toward the sensor from a start position...
6 cm away, stopping at 3 cm, and then moving back to the start position, achieved using an Ultimaker Original+ 3D printer. The tested fabric was placed on top of the sensor and tightened using a plastic frame.

Ten samples were collected for each fabric, with or without the presence of a covering fabric. A 1850 ms window, the movement time of the plate, was used for each sample to calculate the SNR. An average SNR was then calculated for each fabric per condition. The attenuation was calculated using the (logarithm) difference of the SNR with and without the fabric. It represents the ratio of the signal strength between two conditions. In total, I gathered 170 samples and I show the result in Figure 3.33.

Figure 3.32: The box plot of the SNR for each gesture.

Figure 3.33: The attenuation effect of covering fabric.
The result showed that the fabrics woven with metallic thread caused a significant loss of signal (larger than 8db). As these fabrics could effectively block the signal of the sensor, I used them for shielding. Other types of fabric do not cause any significant attenuation of signal strength (within 2.17db). In reference to the results of my main evaluation, the attenuations of these fabrics are all lower than the variation of SNR for each gesture. It means that my model should be capable of handling such variation in the signal caused by the tested fabric. The effect of attenuation may slightly shorten the interaction range but may also not significantly drop the accuracy of my system either.

3.4.9 Limitations and Future Work

In this section, I discuss the limitations of my work and propose potential directions for future research.

Effect of Body Motion

Sensor readings may be different if the antenna is in motion. For example, if a user is jogging, the training data acquired in a stationary condition may be insufficient for recognizing the same set of gestures because the relative motion between the hand and sensor has changed. This is not a problem unique to Fabriccio, as touch input on wearables has the same issue. My next step is to investigate the effect of sensor motion caused by different user activities, identify the issues unique to touchless gesture sensing, and explore practical solutions.

Effect of Antenna Deformation

During my lab study and development of demo applications, I did not observe a noticeable impact on sensing performance when there was a small deformation in the antennas or transmission lines. However, I expect that antenna deformation, especially in a large degree, may eventually cause issues and affect the performance of the system on some of
its applications. I plan to systematically investigate the change in the electric field and ultimately the recognition ability of the touchless gestures caused by antenna deformation. This will allow us to understand the challenges for the proposed sensing technique to be used in real-world scenarios and identify novel solutions to overcome the challenges.

**Sensing Region and Range**

Touchless gestures are required to be performed above the sensor. However, some applications may benefit from an extended ability for gesture sensing at any location on the fabric. The proposed sensor is not designed for this purpose. My future work will continue in this direction. For example, I will develop an antenna array in a grid layout to enlarge the sensing region and even sense the coarse movement of the hand in a 2D space. Also, many challenges still exist, with one being the routing of the transmission lines and the interference of antennas.

**Beyond Gesture Sensing**

Radar technologies have found their way in many HCI applications, such as object recognition for tangible interactions [125] or activity sensing for context-aware applications [126]. Many of these technologies have great potential on daily objects covered or made of interactive fabrics. I will explore ways that can allow us to enable these novel interaction capabilities through ubiquitous textile antennas.

**Energy Consumption**

My current implementation is powered using a battery. While sufficient for a research prototype in an early stage, it is expected that sensors of the future need to be self-sustainable. Textile antennas deliver this promise because they have been used for energy harvesting from radio waves in the environment (e.g., GSM) [110, 111]. It is thus possible to make my technique self-powered and backscattered for sensing and data transformation. I see it
a fruitful direction for future research.

### 3.4.10 Conclusion

In this project, I demonstrated the feasibility of recognizing touchless thumb-tip gestures on interactive fabrics using the Doppler effect. I developed a proof-of-concept prototype using a pair of antennas made of conductive thread sewn onto a fabric substrate. I carefully chose the antenna type, configuration, transmission lines, and operating frequency to balance the complexity of the fabrication process and the sensitivity of the system for touchless hand gestures. I demonstrate through a user study with 10 participants and 12 touchless gestures that my system can achieve a 92.8% cross-validation accuracy, and 85.2% cross-session accuracy. For the subset of seven gestures, the cross-user accuracy can reach 87.6%. My technique provides a useful addition to existing sensing techniques for user input on soft fabrics, primarily based on touch and deformation. This enables a new set of applications on everyday objects that are covered or made of interactive fabrics. I believe my technique may serve as important groundwork for integrating the gestural input into the soft objects in people’s daily life.
Chapter 4

Sensitive Computational Material - Enabling Implicit Inputs

In addition to explicit input, implicit input or contextual awareness is also important for everyday object to be "smart" for enabling nuanced and assistive human-computer interactions [127]. For example, a table that can recognize the objects placed on it and the activities of nearby users could facilitate many smart home, smart workplace, and healthcare applications, such as providing automatic task assessment for in-situ feedback and informing caregivers of patients’ work practices and diets. While many applications can be enabled by context-aware computing, today’s systems are typically oblivious to their physical context and ignorant of humans. To endow everyday objects made of computational materials with contextual awareness, an interface must be created that can recognize implicit inputs and contexts on these computational materials.

In this dissertation, I demonstrate my techniques for creating a textile interface that can identify the objects it comes into contact with [9, 10]. Once again, I chose textiles for this part because they are widely used in making clothing, pockets, furniture, and bags. Furthermore, the methods I developed in these projects can also be applied to other materials, such as wood, to enhance their awareness. In the following sections, I will briefly introduce the
related work in this area and provide an overview of the two research projects that I have developed.

4.1 Background

In contrast to explicit input, implicit input does not require any explicit action from the user to interact with a computer. It is widely used for activity tracking or for contextual interactions. However, there is limited research on implicit input for interactive fabrics compared to explicit input. One such work is eCushion [84]. The technique uses a pressure-sensitive fabric in the form of a seat cushion to infer a user’s seated posture. Another study by Rofouei et al. [86] uses the pressure footprint (weight and shape) to distinguish objects placed on the fabric, but the technique lacks an evaluation to determine its effectiveness. Tessutivo [85] is the most relevant work to my research, but it only works on metallic objects. In contrast, my work focuses on non-metallic objects that are also common in daily life. This research complements existing literature with a new sensing technique based on capacitance.

4.2 Chapter Overview

In this dissertation, I presented two research projects [9, 10] that enabled the implicit inputs for computational textiles. The first project is Capacitivo. It enables a fabric to be capable of sensing a wide variety of non-metallic daily objects it is in contact with. The second project is Project Tasca. It introduces a pocket-based textile sensor that detects user input and recognizes everyday objects that a user carries in the pockets of a pair of pants (e.g., keys, coins, electronic devices, or plastic items). Next, I show the abstract of these research projects to provide readers with an overview of my research projects on this topic.
4.2.1 Chapter 7: Capacitivo

In Capacitivo [9], I present a contact-based object recognition technique developed for interactive fabrics, using capacitive sensing. Unlike prior work that has focused on metallic objects, our technique recognizes non-metallic objects such as food, different types of fruits, liquids, and other types of objects that are often found around a home or in a workplace. To demonstrate my technique, I created a prototype composed of a 12 x 12 grid of electrodes, made from conductive fabric attached to a textile substrate. I designed the size and separation between the electrodes to maximize the sensing area and sensitivity. I then used a 10-person study to evaluate the performance of our sensing technique using 20 different objects, which yielded a 94.5% accuracy rate. I conclude this work by presenting several different application scenarios to demonstrate unique interactions that are enabled by our technique on fabrics.

4.2.2 Chapter 8: Project Tasca

In Project Tasca [10], I present a pocket-based textile sensor that detects user input and recognizes everyday objects that a user carries in the pockets of a pair of pants (e.g., keys, coins, electronic devices, or plastic items). By creating a new fabric-based sensor capable of detecting in-pocket touch and pressure, and recognizing metallic, non-metallic, and tagged objects inside the pocket, I enable a rich variety of subtle, eyes-free, and always-available input, as well as context-driven interactions in wearable scenarios. I developed our prototype by integrating four distinct types of sensing methods, namely: inductive sensing, capacitive sensing, resistive sensing, and NFC in a multi-layer fabric structure into the form factor of a jeans pocket. Through a ten-participant study, I evaluated the performance of our prototype across 11 common objects including hands, 8 force gestures, and 30 NFC tag placements. I yielded a 92.3% personal cross-validation accuracy for object recognition, 96.4% accuracy for gesture recognition, and a 100% accuracy for detecting NFC tags at close distance. I conclude this project by demonstrating the interactions enabled by our
pocket-based sensor in several applications.
4.3 Capacitivo

4.3.1 Introduction

Computational materials have immense potential for enabling smart everyday objects. However, current sensing techniques for computational materials require explicit user actions, such as touching [42, 43] or deforming [44, 45], to interact with fabrics. This limits the computational fabric’s awareness of its context of use, such as the types of objects it comes into contact with. As a result, there are missed opportunities for new applications and interaction techniques.

In this project, I present a contact-based object recognition technique on computational textiles using capacitive sensing. My technique senses and recognizes non-metallic objects that are common in homes or workspaces, such as food items, dinnerware, plastic, and paper products. When an object is in contact with the fabric, my technique recognizes the object based on its capacitance footprint. Consequently, a desired action can be triggered. For example, a smoothie recipe can be suggested to a user based on what fruit or vegetable the user has inside a basket, detected through its cloth lining (Figure 4.1b). Aside from recognizing the contacted object, my system can also sense the change of what is inside a container. For example, a tablecloth can detect whether the soil of a table plant is wet or dry, enabling the system to remind a user to water the plant (Figure 4.1c).

Figure 4.1: (a) Capacitivo is an interactive fabric, capable of sensing a wide variety of non-metallic daily objects it is in contact with. (b) For example, the fabric sensor can sense different types of fruits. (c) It can also sense if the soil of a table plant is wet or dry.
I demonstrate the technical feasibility of my approach through the implementation of a proof-of-concept prototype called Capacitivo. My prototype is composed of a grid of $12 \times 12$ electrodes made of conductive fabric that is attached to a textile substrate (Figure 4.1a). The electrodes are connected by rows and columns, allowing a contacted object to be recognized based on its material as well as the shape of the contact area using mutual and self-capacitive sensing. I carefully designed the size of the electrodes and the distance between them, to maximize the sensitivity of my prototype when placed flat on a tabletop. In a controlled experiment, I tested the recognition accuracy of my approach with 20 daily objects, ranging from food to plastic and dinnerware (empty or filled with water or soup). My results suggested that the prototype could achieve a real-time recognition accuracy of 94.5%.

The contributions of this work include: (1) a non-metallic object recognition technique on interactive fabrics using capacitive sensing; (2) the result of an experiment measuring the accuracy of my technique; and (3) usage scenarios demonstrating unique applications enabled by Capacitivo.

### 4.3.2 Related Work

I briefly discuss the literature for input on fabrics, object recognition using capacitive sensing, and the sensing techniques for context-aware applications.

**Input on Interactive Fabric**

Research on interactive fabrics has been primarily focused on explicit input using touching [42, 43] or deforming getures [44, 45]. The earliest exploration of this space was the Musical Jacket [46], which allows a user to interact with a computer using a fabric-based touch keypad embroidered on a jacket. Jacquard [54] from Levi’s and Google is one of the first commercial products in this space. It allows touch gestures to be carried out on the sleeve cuff of a jacket made of touch sensitive fabric using conductive yarn. Recent work
from Wu, et al. [7] shows that touchless hand gestures, like swipe, are also possible on interactive fabrics using Doppler motion sensing. Aside from touch and touchless gestures, deformation gestures have also been studied for explicit input on fabrics. StretchEBand [65], for example, allows a user to interact with computing devices by stretching, folding, and pressing a soft fabric band made of a textile stretch sensor. SmartSleeve [44] enables even an even wider range of deformation gestures on the sleeve using a pressure-sensitive textile sensor.

In contrast to explicit input, implicit input does not require explicit action from a user to interact with a computer. It is widely used for activity tracking or for contextual interactions. In comparison to explicit input, there is little research on implicit input for interactive fabrics. eCushion [84] is one of them. The technique uses a pressure sensitive fabric developed in the form of a seat cushion to infer a user’s seated posture. Rofouewe et al.’s work [86] uses the pressure footprint (e.g., weight and shape) to distinguish objects placed on the fabric. No evaluation was conducted to inform how well this technique works. Tessutivo [85] is the most relevant work to my research, but the technique only works on metallic objects. In contrast, my work focuses on non-metallic objects that are also common in daily life. It complements existing research in the literature with a new sensing technique based on capacitance.

**Capacitive Sensing for Object Recognition**

Capacitive sensing is a well-known technique that has been used in prior research for numerous applications, including sensing touch input [128, 129, 130, 131, 132], mid-air hand gesture and postures [133, 134, 135, 132], differentiating people [131, 71], sensing the distance or displacement of an object [136, 137], and analyzing the material of an object [133, 138, 139][8, 32, 38, 43]. An overview of capacitive sensing and its applications was described by Grosse-Puppendahl, et al. [140].

For object recognition, a common approach is to tag the target object. For example,
TUIC [141] identifies tagged objects through a commodity touchscreen (i.e. mutual capacitance sensing) by recognizing the geometrical pattern of the tag. With Capacitive NFCs [142], target objects are instrumented using an active tag. Aside from detecting them, the motion of the objects can also be tracked using self-capacitive sensing. Zanzibar [143] uses both mutual and self-capacitive sensing in a single device to sense touch and touchless hand gestures. Object recognition was implemented using NFC tags. While tagging can be an effective approach in some applications (e.g., tangible interaction), usability can be a concern in contextual applications if it is a requirement for a user to tag every object involved in an application.

This has led to numerous techniques being developed to sense untagged objects. One of the more common techniques in this body of research is Electrical Capacitance Tomography (ECT) [138, 144]. The technique measures the distribution of capacitance across sensor electrodes affected by the target object. Unlike the other approaches, where the electrodes can be connected by rows or columns, systems using ECT require each individual electrode to be connected to the sensor board separately. This may increase the complexity of the system. My technique is also based on capacitive sensing, but with a simpler structure suitable for interactive fabrics. In comparison to existing work, my approach is unique in that I combined both mutual and self-capacitance in a single package, to recognize an object based on the material and shape of the contact area of the object.

**Sensing Techniques for Context Awareness**

Aside from capacitance, implicit input can also be sensed using GPS or motion sensors on a smartphone [145], millimeter wave radar [146], computer vision [147], acoustics for objects that emit a sound [148], electromagnetic noises in the environment [149], and contact vibrations when people use their hands [20]. Within existing research, techniques like EM-Sense [149] and ViBand [20] leverage the intrinsic electric or mechanical properties of home appliances (e.g., electromagnetic noises or mechanical vibrations) to infer a user’s
activities and how devices are used by users. Other lines of research in this space focus on the recognition of objects based on the difference in the material of the objects, and thus work with a large variety of passive objects or even liquids. Radarcat [146], for example, is a technique based on millimeter wave radar. It can recognize passive and active objects as well as different parts of the body. Furthermore, TagScan [150] identifies different types of liquids using RFID. Unlike the existing work, where objects can only be detected at a single location of a rigid sensor, my technique is designed for soft fabrics to cover a broader surface area.

4.3.3 Sensing Principle

My object recognition technique is based on sensing the change in the capacitance of electrodes, affected by the presence of an object. For example, when a non-metallic object is in contact with the electrodes, the electric field applied from the electrodes causes an electric displacement within the object. The amount of the electric displacement varies among different objects, depending on the permittivity of the objects. The electric displacement changes the charge stored in the electrodes, and in turn alters the capacitance. It is thus possible to detect or recognize an object based on how much of a shift is observed in the measured capacitance. When a metallic object is in contact with the electrodes, the shift in the measured capacitance is primarily caused by short circuit or the dielectric (e.g. air) in the tiny spaces between the uneven contact surfaces of the object and the electrode. The change in the capacitance is thus not related to the material and cannot be used for recognition of objects.

Unlike sensing gestural input from fingers, object recognition relies on precise sensor readings from different types of objects. Traditional methods, like time-based capacitance measurement, lack precision. I employed a resonance-based approach, where the sensing unit is composed of an LC resonant circuit [33], including an inductor and capacitor (sensor electrodes). By precisely measuring the resonant frequency (f) of the circuit, the
capacitance (C) of the electrodes can be calculated using the following formula:

\[
C = \frac{1}{4\pi^2 f^2 L}
\]

where L is the known inductance. The capacitance measured using this approach is composed of the capacitance of different types occurring in the circuit, primarily the mutual and self-capacitance in my case (details provided later).

Unlike alternative methods, the resonance-based approach has two major advantages that are essential for robust object recognition. First, it is less susceptible to EMI noises, thus having a better signal-noise-ratio (SNR). Second, it allows the capacitance to be measured in a wider range (1pF to 250nF) with an ultra-high resolution (0.08fF) [151].

### 4.3.4 Hardware Implementation

I present the implementation details of Capacitivo using conductive fabrics and customized hardware.

**Sensor Design**

My sensor is composed of coplanar electrodes connected by rows and columns and a ground plane (Figure 4.2). With this setup, two types of capacitance occur primarily: mutual and self-capacitance. Mutual capacitance is the capacitance between the adjacent electrodes while self-capacitance is the capacitance between the electrodes and the ground plane. Both are affected by a contacted object. I used the aggregation of them to allow the signal to be more pronounced in response to the small change in the capacitance. For each capacitance measurement, one row and one column are selected to form a mutual capacitor. The selected electrodes also act as capacitors coupling to the ground, so the changes in both types of capacitance together affect the sensor readings (oscillation frequency). The impact caused by a contacted object can thus be measured by scanning all the row and col-
umn electrodes. However, the challenge with this approach is that due to the coupling of the
electrodes to the ground (self-capacitance), sensor readings at a location are interfered with
all electrodes connected in the same row or column, which I call a crossing effect (Figure
4.3b). This affects the sensing accuracy due to the noise outside the object´s contact area.

Figure 4.2: An illustration of the mutual and self-capacitance formed by my sensor.

One approach in solving this problem is to connect each individual electrode to the
sensor board separately, but it comes at the cost of system complexity and scalability. I
addressed the crossing effect issue by additionally measuring the contour of the contact
area of the object and gathering sensor readings from only inside of the contour. The shape
of the contact area of the object can also be useful for object recognition. With my sensor
setup the shape of the contact area of the object can be roughly captured using mutual
capacitive sensing (Figure 4.3c). The issue, however, is that the current resonance-based
approach cannot measure the mutual capacitance in isolation. I thus had to use a separate
circuit, employing a method commonly used for mutual capacitive sensing by measuring
the displacement current from a transmitter to a receiver (discussed later). In comparison
to the resonant-based approach, this method is less sensitive to the difference in object
material, so it was only used for sensing shape of the contact area of the object.
Figure 4.3: (a) A credit card placed on my textile sensor. (b) The footprint of it measured by the resonance-based approach. (c) The footprint of it measured by the mutual capacitive sensing.

**Fabricating Sensor Electrodes**

I describe the fabrication approach to create the fabric sensor using three layers of conductive and non-conductive (substrate) fabric.

With my approach, electrodes are created using conductive fabric attaching to a layer of non-conductive substrate. I began by attaching conductive fabric (I used Adafruit conductive woven fabric) to a cotton substrate using an iron-on adhesive. After the adhesive dried, I used a low-cost cutting machine (Cricut Air Explorer) to cut the conductive layer into diamond shaped electrodes (Figure 4.4a). In principle, the electrodes can be made in any shape. I used the diamond shape to maximize the sensing region in a 2D space. I carefully adjusted the cutting force of the cutting machine so that it only cuts the conductive layer without damaging the substrate. Once the cut was completed, I heated the fabric
Figure 4.4: Fabrication process of sensor electrodes. (a) The electrodes are being cut using a low-cost cutting machine. b) Unwanted piece is being peeled off the substrate. (c) Connecting the column electrodes from the back using a sewing machine. (d) A grounded shielding layer is being attached to the back side of the sensor.

again to soften the adhesive to allow the unwanted piece to be peeled off the substrate (Figure 4.4b). I designed the electrode pattern using graphics programming software (e.g., Processing) and then saved into an SVG file that was readable by the cutting machine. In comparison to alternative methods like embroidering [152], knitting [153], weaving [154], my approach creates electrodes that are precise in shape and location, while keeping the costs low.

For the electrodes on rows, I also cut the connection lines between them. Column electrodes were connected from the back using conductive threads so that they are electrically disconnected from the row electrodes. This can be done by hand or using a home sewing machine (Figure 4.4c). When using a sewing machine, the standard stitching process pushes the threads through the substrate and electrode layer. This connects the front and back of the sensor, which is needed to route the column connection lines to the back. The issue, however, is that the conductive threads left on the front may cross the connection lines of the row electrodes, which causes short circuits between the rows and columns.

I solved this problem by controlling when the conductive thread could be pushed to the front side of the sensor. This was done by fine-tuning the tension and speed of the sewing machine. I first adopted the method discussed in Dunne et. al.’s work [123], and carefully tuned the tension of the top thread (e.g., non-conductive thread) to ensure that when the machine is sewing at a high speed, the conductive thread (LIBERATOR 40) on the back
only floated on the surface of the substrate without penetrating it. Adjusting the machine to sew at a low speed allowed the conductive thread to be pushed through and land on the front side of the sensor (e.g., connecting to an electrode). To force the machine to sew at a low speed, I let it turn at a sharp angle (90°) at the location where the penetration of the conductive thread was needed. Like similar products in the consumer market, my sewing machine sews at a low speed at the corners.

Finally, I added a grounded shielding layer (i.e. ground plane) made of a knit conductive fabric (Adafruit knit jersey conductive fabric) on the back side of the sensor (Figure 4.4d). The knit conductive fabric is attached to the back of the sensor on its non-conductive side to avoid shorting the column electrodes. An optional layer of fabric can be used to cover the electrodes for the aesthetic sake. My prototype employed a 12×12 grid layout of electrodes (15.6 × 15.6 cm), taking 20 minutes to complete the fabrication process, with a total material cost under $30 USD.

**Customized Sensing Board**

My customized sensing board (Figure 4.5) uses an LC circuit with a FDC2214 chip from Texas Instruments. The FDC2214 chip measures the joint effect of mutual and self-capacitance. To capture the contour of the contact area of an object and to mitigate the crossing effect, I used a separate mutual capacitive sensing circuit, similar to the one described in prior work [155]. On the transmitter side (column electrodes), the circuit is composed of a wave generator (AD5930, Analog Device) and an amplifier (AD8066ARZ, Analog Device), which generates an excitation signal using a 100k Hz sine wave with a 5V peak-to-peak voltage. The excitation signal is routed to the row electrodes via multiplexers. On the receiver side (row electrodes), the displacement current from the column electrodes was converted and filtered to an amplified voltage signal using two amplifiers. The system then reports the capacitance by calculating the RMS value of the voltage signal using a window size of 100 samples with a sampling rate of 1MHz.
Aside from the two capacitive sensing circuits, my sensing board also has an ARM-based flash micro-controller (ATSAM3X8EA-AU) powered by Arduino Due Firmware, a Bluetooth module (RN42, Microchip) for wireless communication, and eight 4:1 multiplexers (FSUSB74, ON Semiconductor) to drive the electrode arrays. Each multiplexer further connects to two switches (TMUX154EDGSR, Texas Instruments) to allow the system to switch between the two capacitive sensing circuits. Note that I used two switches instead of one, because I found that the sensor readings interfered with each other if the two circuits are connected to the same switch.

Note that when the system is in one of the two capacitive sensing modes (e.g., resonance-based or mutual), the circuit for the other mode is turned off to avoid crosstalk. With my current implementation, the prototype runs at 3 - 4 Hz for the 12×12 layout of electrodes. The speed bottleneck is in my mutual capacitive sensing. But this issue can be mitigated in the future using better sensing chips (e.g. MTCH6303 [21]).

Figure 4.5: Capacitive sensing board. The button and header pins are used for debugging.
Wire Connection

The sensor board is connected to the fabric electrodes through electric wires and conductive threads. I used a method similar to the one suggested in prior research [85] to connect the lead threads and wires together using twist splice. I used hot glue to strengthen and fix the connection. Although it is clunky, this type of wire connection is stable and performed well in my experiments.

4.3.5 Determining Electrode Size and Separation

For a given input voltage (e.g., 5V in my case), the size of the electrodes and the distance between two adjacent electrodes (called separation as shown in Figure 4.6), may affect the intensity of the electric field [156], thus affecting sensor sensitivity to the small changes in the capacitance caused by objects of different materials. In theory, increasing the size of the electrodes increases the intensity of the electric field. The tradeoff, however, is that larger electrodes reduce the sensing resolution in 2D. The effect of separation adds more complexity. For example, reducing the separation may increase the intensity of the electric field for mutual capacitance since the electrodes become closer. However, an opposite trend may be observed for self-capacitance since the space between the electrodes becomes smaller. I conducted two experiments to determine the proper size and separation for my implementation.

Electrode Size

For the study about electrode size, I created three sensors with different electrode sizes (7mm, 14mm, and 21 mm in diagonal distance). 7mm was chosen to be the smallest size for my study because it was nearly the smallest electrode that could be robustly fabricated using the low-cost cutting machines that are widely available to the research and maker communities. The electrodes of the sensors were made in a 2×2 grid with a fixed 2 mm
separation. For each electrode size, I also created five replications of the same sensor for data collection, to accommodate any potential variations in sensor readings caused by inconsistency in fabrication.

![Figure 4.6: An illustration of electrode size and separation.](image)

To acquire the sensor data for comparison, I used a thin plastic sheet as a dummy object (60 mm × 60 mm × 1 mm), 3D printed using PLA filament. I chose PLA because it has a relatively low permittivity (e.g., 3.5 F/m), allowing us to better measure the size effect of the electrodes in reaction to a small change in capacitance. For each trial, I placed the plastic sheet on top of the sensor and covered all electrodes. I measured the sensor readings for each column-row pair of the electrodes to calculate a SNR. The average SNR was then calculated for all the pairs of the sensor and then all the five replications of the same sensor.

**Results:** As I expected, increasing the size of the electrodes increases SNR. The SNR achieved from size 21mm is nearly five times as high as that achieved from size 7mm. The tradeoff is of course sensing resolution in 2D. To consider both SNR and 2D resolution, I calculated an overall score for each electrode size, using the following equation. The higher the score is, the better the size is for the electrode to satisfy my sensing needs.

\[
Score = \frac{SNR}{SNR_{highest}} + \frac{Resolution\text{Score}}{Resolution\text{Score}_{highest}}
\]
where ResolutionScore represents the number of electrodes per square meter. The result is shown in Figure 4.7. Among all the tested electrode sizes, 7 mm scored the highest despite being the lowest on SNR. I thus used 7 mm in my implementation.

**Electrode Separation**

For the separation study, I created three sensors each with different separation distances (2mm, 3mm, and 4 mm). The sensors were created using 7mm electrodes. 2mm was chosen to be the smallest separation for my study because again it was approximately the smallest separation that can be robustly fabricated using the low-cost cutting machines. Similar to the size study, I also created five replications of the same sensor for data collection. All other study protocols remained the same.

**Results**

Interestingly, increasing the distance between two adjacent electrodes also increases SNR. This suggests that self-capacitance may have dominated the sensor signal. Regardless, using the joint effect from both types of capacitance leads to higher SNR. The overall scores shown in Figure 4.7 suggest that the separation of 4 mm has the highest score amongst the tested values. It increases SNR for about twice as high without significantly impacting 2D resolution. I thus used the 4mm separation in my implementation. Note that it was not my goal to identify the most optimal size or separation, thus I left them for future research.

**4.3.6 Object recognition**

My system recognizes non-metallic objects based on the capacitance footprint of the object’s contact area. This section presents my object recognition pipeline.
Figure 4.7: The results of my size and separation experiment. The columns shaded in grey were used in my implementation.

Data Processing

When an object is in contact with the fabric sensor, the sensor reports two 12 × 12 arrays of capacitance values, one from the LC circuit (more precise for sensing material) and the other from the mutual capacitive sensing circuit (more precise for sensing shape). Before the raw sensor data was used for object recognition, I subtracted background noise from the sensor readings for all the column and row electrode pairs. To handle this in an efficient way, I created a 2D noise profile, which represents the background noise across the entire sensor. The noise profile was constructed by averaging the sensor readings at all locations within a sliding window of size 5. It was updated automatically when no object was detected and when no location reported a delta in sensor value (current value minus initial value) exceeding a preset threshold (e.g., 3000 units of FDC2214’s raw sensor reading).

When an object was in contact with the sensor, my system detected the presence of the object if the average of the sensor data across all the locations is over a threshold (5000) while the standard deviation is below a threshold (1000). I then extracted the 2D capacitive footprint of the object by taking the LC circuit data from the contact area of the object. The contact area of the object is determined by getting the locations, whose sensor data is above
30% of the maximum value of the mutual capacitive sensor. The footprint data was then smoothed using a median filter to reduce the fluctuations in sensor readings. In addition, by placing the objects on the sensor one by one, the footprints of multiple objects can be detected. To get a high-resolution visualization of the counter image, I multiplied the data from the LC circuits with the normalized data from the mutual capacitive sensing at each location and scaled up the resulting data map using linear interpretation.

**Machine Learning**

Object recognition was carried out using the Random Forest technique. In comparison to the alternative methods (e.g., Hidden Markov Models and Convolutional Neural Networks), Random Forest is accurate, robust, and more efficient in computation, thus suitable for real-time applications in low-power embedded platforms. Features used for training and testing are primarily based on two types of information, the material and shape of the contact area of the objects, where the shape data was acquired using the interpolated image of the data map. Based on my observation and initial tests, I derived 33 material-related features and 53 shape-related features (Table 4.1).

### 4.3.7 System Evaluation

I conducted an experiment to measure the recognition accuracy of my prototype for daily objects commonly found at home or in an office environment.

**Participants**

Ten right-handed participants (average age: 22.9, 6 males, 4 females) were recruited for the study to reduce tested objects being placed at the same locations or orientations.
Objects

I tested my prototype using 20 everyday objects (Figure 4.8), ranging from food, to personal items and things that are commonly seen in kitchens and offices. The tested objects vary in geometrical (e.g., size, shape) and material properties, which is useful for demonstrating the capability of my sensing technique. I also purposefully included container-like objects, such as a water glass and bowl, to test how reliable it is for my system to recognize the different statuses of containers. For example, I tested how well it could be recognized if a container was empty versus when it was filled with water (e.g., water glass filled with

<table>
<thead>
<tr>
<th>Shape-related Features (53)</th>
<th>Material-related Features or Pressure-related Features (33)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Local Binary Pattern (36)</td>
<td>• Statistical Functions (13): Sum(1), Mean(1), Max(1), Binned Entropy (1), Local Maximum Numbers (1), Median (1), Quantiles (3), Count above/below mean (2), Variance(1), Absolute energy of the object’s pixel values (1)</td>
</tr>
<tr>
<td>• Hu Moments (7)</td>
<td>• Ten-Fold Stats (20): Sort and divide the object’s pixel values into 10 folds and average for each fold (10), Divide grayscale values (e.g., 0 255) into ten intervals and count the number of the pixels in each interval (10)</td>
</tr>
<tr>
<td>• Object Area (1): Number of pixels the object covers</td>
<td></td>
</tr>
<tr>
<td>• Object Edge (1): Number of pixels on object’s edge</td>
<td></td>
</tr>
<tr>
<td>• Average Distance (4): Average distance from object’s pixels to object’s center of gravity and geometric center (2), average distance from object’s edge pixels to object’s center of gravity and geometric center (2)</td>
<td></td>
</tr>
<tr>
<td>• Object’s center of gravity and geometric center(4)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: The feature set extracted from each sensor data for training my machine learning model.
water) or soup (e.g., bowl filled with clam chowder). Similarly, I tested if the system can correctly differentiate between dry and wet soil for a small tabletop plant. This is to show the potential of my system in wider application scenarios. For food items, I bought two packs of the same type of food from different markets or brands. I used those from one pack for training and the other pack for testing. For example, I used cheese from Velveeta for training and cheese from Kraft for testing. I used 3 kiwis, 3 avocados, and 3 grapefruit from one pack for training and same amount of them from another pack for testing.

Data Collection

My study protocol was similar to that of prior work [85]. Three days before the study, a volunteer was invited to collect the training data for the machine learning model. The volunteer was asked to place each of the tested objects inside the sensing area in a random location, orientation, and order. No other instruction was given in terms of how the objects should be presented to the sensor. Fifty samples were collected for each object to train the model. During data collection, the sensor was placed in a horizontal position on a table to mimic a common tablecloth scenario. The device was powered by a wall outlet (earth ground). The same procedure was used for collecting the testing data, except that 10 new participants were recruited for the task. Ten samples were collected per object for testing. Real-time recognition accuracy was recorded for analyzing the study results.
Results

Overall, my system yields an accuracy of 94.5% (SD = 4.5). The confusion matrix of the study result shows that 18 out of the 20 tested objects achieved an accuracy higher than 90% (Figure 4.9). This is a promising result considering that many of the tested objects are similar in shape, and in a few cases, also in material. For example, the Empty Water Glass, Empty Bowl, and Table Salt are all round in their contact area, but they could be recognized by the system with a relatively high accuracy. While the JCPenney Rewards Card scored lower than 90%, it was still differentiable from the Discover Credit Card, which is almost identical in shape and size. Since the two cards are both made of plastic, they are differentiable by the system through the difference in internal structure (e.g., with vs without the chip) and magnetic strip.

The system also performed relatively well on food items. For example, the differences between the Kiwis and Avocados could be discerned with good accuracy despite their size and shape are not very much different from each other. Grapefruits have a unique capacitive footprint among all the tested objects, which allowed them to be recognized with a good accuracy of 98%. In addition to the objects, I found that the status of the container could also be well tracked. For example, the system could distinguish between the Empty Water Glass and the one filled with water with an accuracy of 94% and higher. It could also correctly recognize the Empty Bowl versus the one filled with clam chowder in all the 20 tested instances. Finally, the system was able to differentiate between wet and dry soil for the table plant with 99% accuracy. This is a promising result considering that all the containers, despite their status, are well recognizable among all the other tested objects.

Misclassifications typically occurred when an object failed to have a clear capacitance footprint in my current implementation, such as those with low permittivity (e.g., Credit Cards, Book). The JCPenney Rewards Card is an example, which is one of the two most difficult objects to recognize (85% accuracy). The problem was more pronounced when a part of the card lost contact with the sensor electrodes, which is a common scenario since
the fabric sensor is not entirely flat. In this case, neither the shape nor the material could serve as a reliable indicator for the system to recognize the object. This is partially why the system can get confused between the JCPenney Rewards Card and the Apple AirPods Case.

### 4.3.8 Supplementary Studies

Aside from the main study, I conducted two supplementary studies to push the limit of this fabric sensor further. I first preliminarily evaluated how well the system can recognize different types of liquids. I then investigated how well the objects can be recognized by the sensor in a vertical placement (e.g., in a pocket form factor) to mimic another common
use scenario. These two studies were carried out with a single participant (female, right-handed, 25 years old).

**Liquid Recognition**

To understand how well the proposed technique may work for a wider variety of drinks, I extended my apparatus to six liquids, including Cold Water, Hot Water, Coke, Apple Cider, Milk, and Beer (Pabst blue ribbon). According to prior work [3, 37], the permittivity of these liquids varies due to the difference in temperature and also in the concentration of sugar and salt. However, unlike the objects tested in the main study, liquids are different in a much more subtle way. The result of my initial testing revealed poor performance of my system on the tested liquids, primarily attributed to the small inconsistency in the sensor readings at different locations of the current prototype. I thus only report the performance of the system measured at a fixed location, picked randomly inside the sensor (Figure 4.10 right). While this setup clearly limits the utility of the system, some applications may still benefit from it, including the one related to the smart coaster described in the demo section. I see that this inconsistency issue can likely be resolved by using an industrial grade fabrication process.

**Data Collection:** The study had two sessions, one for training and one for testing. For the training session, data was collected by the participant putting the glass filled with one of the tested liquids anywhere inside a sensing region, made of $7 \times 7$ electrodes. All the liquids were at room temperature ($23 \, ^\circ C$) except the hot water, which was measured at $80 \, ^\circ C$. Aside from the liquids, I also included an empty glass in this study, resulting in a total of 8 conditions. The liquids were tested in a random order. I collected 20 samples for each type of liquid. The same procedure was used in the testing session, which took place six hours later.

**Results:** The study result was analyzed using a twofold cross-validation. Overall, the system achieved an average accuracy of 90.71% (SD=14.6). Figure 4.10 (left) shows
the confusion matrix of the result across all the tested liquids, amongst which, Beer had the lowest accuracy of 65%. It was confused by the system with several other liquids, such as Coke, Apple Cider, and Milk. This is likely because of the similarity in sugar concentration, but it must be confirmed with a careful study. The recognition accuracy increased to 96.67% (SD=6.0) after I removed the Beer from the tested objects.

<table>
<thead>
<tr>
<th></th>
<th>Empty</th>
<th>Cold Water</th>
<th>Hot Water</th>
<th>Coke</th>
<th>Apple Cider</th>
<th>Milk</th>
<th>Beer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cold Water</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hot Water</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Coke</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Apple Cider</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>75</td>
<td>0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Milk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Beer</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>65</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.10: Left: the confusion matrix of the study result. Right: the sensor at a fixed location was used for this study.

**Vertical Sensor Placement**

The goal of the second preliminary study was to measure the performance of the sensor in a vertical placement. The study apparatus included the proposed fabric sensor implemented in a pocket form factor (15.6 × 15.6 cm; Figure 4.11 right), similar to the ones on used on winter jackets. When in a vertical position, the sensor deformed slightly due to gravity. I left it as is to simulate more realistic scenarios.

**Data Collection:** Out of the 20 objects tested in the main study, nine can be fit into the pocket prototype. These include an AirPods Case, Lipstick, Kiwis, Avocados, Cheese Slices in Plastic Wrap, a Discover Credit Card, a JCPenny Rewards Card, a Book, and a
Portable External Hard Drive. I used all except the JCPenny card because its footprint was too weak to be recognized by the pocket prototype. Data was also collected in two sessions with one for training and the other for testing. Similar to the first preliminary study, the two sessions were set 6 hours apart. In each session, the volunteer randomly chose an object and placed it inside the pocket. No instruction was given regarding how the item should be placed inside the pocket in terms of location, or orientation. I collected 20 samples per item for training and testing.

**Result:** The study result was analyzed using a twofold cross-validation. On average, the system could recognize the eight tested objects at an accuracy of 70% (SD = 15.5). That is nearly 25% lower than the accuracy achievable with even more items when the sensor is in the horizontal placement. This is understandable because my sensing technique is contact based. When the sensor is placed vertically, the contact between the object and the sensor cannot be guaranteed since the object may partially lean against the sensor or due to the deformation of the sensor itself.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>E</th>
<th>F</th>
<th>M</th>
<th>R</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>70</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>65</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>75</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>80</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
<td>5</td>
<td>15</td>
<td>0</td>
<td>10</td>
<td>70</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>15</td>
<td>10</td>
<td>25</td>
<td>5</td>
<td>0</td>
<td>35</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>85</td>
</tr>
</tbody>
</table>

Figure 4.11: Left: the confusion matrix of the study result. Right: the sensor in a pocket form factor

This is because the contact area of the objects varied randomly depending on how they were placed and leaned against the sensor inside the pocket. If I removed the Book and
Lipstick from the dataset, the recognition accuracy increased to 80% (SD = 6.3). While Avocados were confused by the system with Kiwis for 10% of the tested instances, the accuracy was not worse than when the sensor was in the horizontal position. If Avocados were removed from the dataset, the system could recognize the remaining objects (e.g., AirPods Case, Kiwis, Cheese Slices in Plastic Wrap, Discover Credit Card, and Portable External Hard Drive) at an accuracy of 87% (SD = 6.7). I think this is encouraging because the result suggests the potential of the sensor in a vertical placement in applications suitable for a small set of objects.

4.3.9 Demo Applications

In this section, I present eight usage scenarios of Capacitivo to demonstrate new applications enabled.

Figure 4.12: The demo applications of Capacitivo. (a) A user can be notified that they’ve left their airpods inside their jacket pocket; (b) A tablecloth can remind a user to not forget a personal item through a smart speaker; (c) Credit card information can be auto filled while shopping, by placing it on a tablecloth; (d) Guided cooking instructions by detecting different food items; (e) A user can track the beverage they are consuming; (f) A user can be gently reminded to clean up after a meal.

Small devices, like AirPods, have become an important part of the ecosystem of the wearables people use in everyday life. However, despite being convenient to carry and use,
their small and portable nature has made them easy to lose and hard to find. For example, people often forget where they leave their AirPods case when the headphones need to be charged. I implemented an interactive pocket on a jacket, which senses if the case is left inside the pocket (Figure 4.12-a). The user can thus be notified where to find it when the battery of the headphones is low.

At home or at a workspace, a table is another location where personal or shared items can be found. I implemented my sensor on a tablecloth, which detects the items that are placed on the table. My system uses a smart speaker to remind a user if something important is left on the table for too long (e.g., credit card) or if they forget to take it before their day begins. In my implementation, I remind the user to take personal items (e.g., lipstick in my example) that are often forgotten when rushing out in the morning (Figure 4.12-b).

When a user registers or shops at a website for the first time, the tablecloth recognizes the credit card placed on the table and automatically fills in the card information for the user (Figure 4.12-c). This is convenient for some users, as entering a card number can be error prone and tedious. Further, the tablecloth can detect if the user's table plant needs watering, reminding a user when the soil is dry.

Capacitivo can also be useful in a kitchen for cooking and eating. For example, cooking is a process which often relies on following a procedure. Thus, a smart kitchen capable of detecting the ingredients and seasonings available on a table, can be helpful for informing a user about the order and timing in which ingredients should be added to a dish that is being prepared. My implementation can detect avocado, cheese, and salt on the table and provide a user with suggestions on how to make avocado soup (Figure 4.12-d).

Sometimes cooking also needs spontaneity. When the user does not know what to make for dinner or to drink, it can be helpful to suggest the user recipes based on what is available at home. In my implementation, the system can suggest to the user a smoothie recipe based on what fruit and vegetables they have in a basket (e.g., kiwis, avocados, grapefruits), detectable by the cloth lining of the basket. During dinner, the system infers what a user
drinks based on sensing the liquid inside the glass. It automatically updates their diet tracking app (Figure 4.12-e). Finally, the system can remind the user to clean the table after the empty bowl used for soup was left on the table for several hours (Figure 4.12-f).

4.3.10 Limitations and Future Work

I discuss the insights gained from this work, the limitations of my approach, and future research directions.

Sensor Deformation

My proposed fabric sensor is deformable, making it suitable for objects or containers with a curved surface (e.g., basket). To accommodate this, the machine learning model needs to be trained with the sensor in the corresponding curved form factor. This allows the sensor to be used in a wide variety of different application scenarios. However, the problem with the current implementation is that if the electrodes are deformed after the machine learning model is trained, sensor readings are affected, which consequently introduces false recognition. This is an issue with many of the existing fabric sensors and requires careful research in the future. I am investigating methods that can model the shape deformation of the sensor using the unique sensor readings introduced by folding the electrodes at different locations, degrees, and angles. However, many of the application scenarios foreseeable for the proposed sensor, (e.g. those in this paper) allow the sensor to work under relatively stable conditions without being prone to significant deformation.

Touch Input

As my technique uses capacitive sensing, it is expected that finger touch gestures can be sensed. This enables new applications on top of what are proposed here in the paper. I left this part of research for future work because my prototype in its current form was developed with a focus on object recognition. Sensing the movement of the finger or a detected object
requires a higher frame rate from a system than what is developed in the current prototype (3 – 4 Hz). As mentioned earlier, the frame rate of my system is currently limited by the chip used for mutual capacitive sensing. It can be increased with better hardware.

**Contact-Based Sensing**

My technique requires the full contact area of the object to be presented to the sensor for the object to be correctly recognized. In some scenarios, a firm contact may not be guaranteed as shown in my preliminary study with the sensor in the vertical placement. This largely impacts sensing accuracy. A potential solution to this problem is to optimize sensitivity also in the Z direction so that objects can be sensed in the near field of the electrodes. Fabrication. With my current implementation, sensor readings vary slightly at different locations of the electrode array. This made it difficult for the system to sense small differences between the impact in the capacitance caused by the different types of liquids. Future work will focus on improving the fabrication process to assure the consistency in sensor readings across the surface of the sensor. Solving this means recognition accuracy of the system over other types of daily objects can be further improved.

**Sensing Metallic Objects**

My technique does not work with metallic objects. This is a limitation of capacitive sensing. I expect that a hybrid approach combining capacitive sensing with the other types of sensing techniques, such as inductive sensing, has the potential to solve this problem. This may require changing the geometry of the electrodes to accommodate both types of sensing.

**4.3.11 Conclusion**

This project demonstrates a technique for contact-based object recognition on interactive fabrics, enabling them to detect everyday objects. I first discussed my sensing principle and
investigations into the size and separation between electrodes that were optimal to enable my approach. I then built a prototype with a 12 x 12 grid of electrodes made from conductive fabrics, which was chosen based on my earlier investigations. Using a ten-participant study, I found my approach demonstrated a 94.5% real time classification accuracy with 20 different objects, enabling a new set of application scenarios. I believe my approach is a critical step for increasing the input space of interactive fabrics.
4.4 Project Tasca

4.4.1 Introduction

Concepts of smart pockets (e.g., on a pair of pants) allow touch interactions to not only occur beyond smartphones, watches or rings, but also be carried out in a comfortable, private, and always-available manner to use other computing devices (e.g., head-worn or wall-size displays [157, 158]). Furthermore, understanding the items that a user has in a pocket (e.g., a phone) also enables a new set of applications, such as activity tracking [159], placement-dependent notification [160], or providing new context to the information sensed from other devices [160].

While previous research has explored the concept of smart pockets, much of the focus was demonstrating proof-of-concepts and application scenarios using mockups through rigid devices or sensors, such as a camera [158], optical sensor panel [159], or capacitive touch panel [157], which are unsuitable for practical and daily use. Therefore, there is a need for technology that enables computational textiles to be used to create smart pockets that can be used practically and daily.

In this project, I introduce Tasca, an interactive pocket-based textile sensor integrated into a form factor that fits into the pocket of a pair of jeans (Figure 4.13). Tasca is capable of sensing a wide variety of user input that exists in the current literature and beyond. Unlike the previous work that I build upon, my prototype was developed using an interactive fabric, and can sense touch [42], pressure [44], while also recognizing everyday items that people carry in their pocket such as keys, coins, electronic devices and some plastic items [159]. I developed my prototype using four different sensing techniques that have not been previously developed into a single fabric-based package: NFC, capacitive, inductive, and resistive sensing. These techniques were combined together to deliver a more practical solution for interactive fabrics in a wearable pocket context, in terms of robustness against sensor deformation (e.g. improved object recognition by fusing data from any pair of the
Figure 4.13: Tasca is a pocketed-based textile sensor, capable of sensing (A) common objects people carry in their pockets (e.g. a smartphone or car key), (B) touch and pressure gestures, and (C) NFC tags.

sensors) and more capable in sensed object types (e.g. metallic, nonmetallic, and NFC tagged objects).

The contributions of my work include: (1) the sensor design of an interactive pants pocket, capable of sensing explicit user input and recognizing the objects a user carries in their pocket and (2) the result of an experimental evaluation of the sensing performance of my system.

4.4.2 Related Work

I briefly discuss the literature for interacting with fabrics, the different sensing techniques for smart fabrics, and the area of smart pockets.

Interacting with Smart Fabrics

Smart fabrics use a number of sensing methods that enable interactions for users, including inductive sensing [85], capacitive touch and gestures, all of which can accomplished using different weaving, braiding, sewing and embroidery techniques [161, 162, 163]. Like other
forms of input, smart fabric input can be broken into implicit and explicit input. Implicit input is often used for contextual interactions (e.g. activity tracking or adjusting an environment), and doesn’t require an explicit action from a user. Examples of such research include using the pressure footprint of an object to detect different objects [164], using a pressure sensitive fabric approach [165]. For smart fabric interactions, implicit input has not been widely explored other than the work such as [85, 9].

Unlike implicit input on fabrics, explicit input techniques and their enabling technologies continue to be well explored. The most common technique for explicit input on smart fabrics is touch [42, 43] or manipulating the fabric itself (deformation) [44, 45]. The canonical example of explicit input on a textile is the Musical Jacket, where a fabric-based keypad was embroidered onto a jacket and allowed a user to play music. More recent examples include Project Jacquard [42] and GestureSleeve [47], both use touch gestures on different parts of a garment. Beyond touch based gestures, mid-air gestures also have been explored by Wu et al. [7], where Dopper motion sensing was integrated into fabric to enable different interactions. For deformation-based input techniques, SmartSleeve demonstrates how common fabric-based interactions such as folding, stretching and pressing can be augmented as a means of interacting with everyday objects [44].

While many of the input techniques for smart fabrics have been explored in individual prototypes, many have not been combined together, providing an interesting opportunity to explore novel implicit and explicit input techniques.

**Sensing Techniques for Smart Fabrics**

A number of techniques have been developed in the research literature for sensing using fabrics, with the most common enabling approach being the combination of fabrics and different sensing coils. Example application scenarios for fabrics and coils include include wireless power charging, and inductive heart sensors [166, 167, 168, 169].

Beyond using solely coils for sensing on smart fabrics, sensing techniques for textiles
have included capacitive and NFC sensing, and object recognition, to name a few. Capacitive sensing involves particular challenges for fabrics and wearables [170], but ultimately it can enable touch input, hand gesture and posture detection (e.g. detecting large swipe gestures with an entire hand) [42], and material analysis [9]. A canonical example of capacitive sensing and textiles that is commercially available, is the Levi’s Jacquard jacket [42], where capacitive sensing is integrated into a jacket cuff and touch gestures are used to interact with a phone and other devices.

Combining different sensing approaches also leads to novel interactions. For example, Project Zanzibar [143] combines mutual and self-capacitive sensing to detect touch and hand gestures, and also used NFC tagging to detect objects. In my work, I also combine different sensing techniques, where I explored the fusion of inductive, capacitive, resistive, and NFC sensing to enable a wide variety of applications new to interactive pockets on a pair of jeans.

**Smart Pocket Interactions**

While the broader space of interactions as it relates to smart garments, and interactions on and around the body is well explored, areas located around the the thigh – specifically the pocket – have been under-explored comparatively [157, 158]. Prior work has demonstrated that in scenarios involving standing, sitting or kneeling, the front of the thigh is the most appropriate position to place a touchpad-like interface [171]. Similarly, Holleis et al. [172] used capacitive buttons integrated into different garment form factors to demonstrate that the thigh area was potentially the most acceptable for touch-based wearable controls.

One critical factor for acceptable interactions around the pocket (and thigh area) is their social acceptability. For example, prior research has shown that people are comfortable with interactions above the belt, but not near the area around a belt buckle, due to social statements that could be perceived (e.g. a hand near the lower extremities) [173]. Profita et al. [174] has also demonstrated similar results that pockets are not as socially acceptable
because of the physical location to different (private) parts of the body. This means that ideally, sociably acceptable interactions should be closer to a resting hand position for pockets (similar to [157]).

An early example of pocket-based interaction in the literature is PocketTouch [175], which demonstrated capacitive sensing through fabric. Their approach consisted of a capacitive sensing grid (connected to a smart phone) that enabled touch interaction through the pocket, as well as stroke-based gestures that could be performed on the outside of the pocket. Through-pocket techniques have also been demonstrated by others [176, 177] and often rely on using the sensors of a mobile phone in a pocket [157]. Smart pocket prototypes have also been created using cameras, and other more rigid materials, but aren’t practical for an everyday pocket. The closest system that recognizes objects in a pocket was created by Shimozuru et al. [178], using an array of infrared sensors, but is also impractical due to its rigidness.

In this work, I am the first to explore how multiple sensing techniques can be designed and developed in combination (rather than in isolation), to enable different types of interactions in an everyday pocket form factor. As part of this exploration, I identify unique challenges due to the constraints of a pocket and demonstrate promising solutions that are both applicable to pockets, as well as the wider space of smart fabrics.

4.4.3 Tasca

My goal was to implement Tasca as a pocket sensor that supports some of the most common input modalities, including: (1) 2D touch gestures commonly used on mobile and wearable devices and (2) force touch by pressing the fabric at different levels of pressure. Additionally, to allow for rich contextual interactions, I wanted the sensor to be able to (3) sense and recognize daily objects that users normally carry in their pants pockets. Finally, to allow customization and enable the system to handle objects that are not registered in my system, I wanted the sensor to be able to (4) sense tags that are easy to attach to objects.
Sensing Capabilities

My prototype is capable of sensing touch, pressure, metallic objects, non-metallic objects, and tagged objects by integrating four different types of sensing methods, including inductive sensing, capacitive sensing, resistive sensing, and NFC. I investigated the sensing techniques that are contact-based or the ones working in a short range so the operation of my sensor does not interfere with other personal electronic devices.

**Sensing Metallic Objects Using Inductive Sensing:** Metallic objects are recognized using inductive sensing based on Faraday’s law of induction. When an alternating electrical current is flowing through a L-C resonator, composed of the spiral-shaped coil of the sensor (inductor) and a capacitor, an electromagnetic field is generated around the sensor. If a conductive object is brought into the vicinity of the sensor, the electromagnetic field will induce an eddy current on the surface of the object, which in turn generates its own electromagnetic field, which opposes the original field generated by the sensor. Therefore, a small shift in inductance can be observed through the sensor. The amount of the shift is related to the resistivity, size, and shape of the object when it is in proximity to the sensor. Inductive sensing works primarily with metallic objects (e.g., keys, coins) and those composed of metallic parts (e.g., electronic devices). To implement inductive sensing for my pocket, I used an approach similar to one described in the research literature [85]. However, the design of my coils is different because I also use the coils for NFC, which I discuss later.

**Sensing Touch and Non-Metallic Objects Using Capacitive Sensing:** Touch input is sensed using capacitive sensing, which is a well-known technique used on everyday devices ranging from smartphones and watches to interactive garments (e.g., Jacquard [42]). Aside from touch input, capacitive sensing has also been used for object recognition [142, 141]. Unlike inductive sensing, capacitive sensing works better for non-metallic objects, such as...
food items, dinnerware, plastic, and paper products. As a complement for inductive sensing, I included capacitive sensing to recognize non-metallic objects (e.g. hand sanitizer and wallet), as well as sensing touch input, using a shared set of coplanar textile electrodes. For object recognition, I used a technique similar to the one described in the research literature [9]. My system recognizes non-metallic objects based on their capacitance footprint introduced by the change in the capacitance of electrodes, affected by the presence of an object. When the electrodes are in contact with a non-metallic object, the electric field applied from the electrodes causes a certain amount of electric displacement within the object. Objects with different permittivity have different effects on the amount of the electric displacement, which alters the capacitance of the object. The shift in the capacitance can be measured using a resonance-based approach, which is known to be precise and less susceptible to environmental noise.

**Sensing Pressure Using Resistive Sensing:** Pressure sensing is based on the change in the resistance of a piezo-resistive material when it is pressed or deformed. As an input method, this resistive sensing can be used for both sensing touch input [44] and recognizing objects [86]. For object recognition, unlike capacitive and inductive sensing, resistive sensing detects objects primarily based on the shape and amount of pressure exerted on the sensor by the objects. In the context of a pocket, using resistive sensing allows my system to infer the thickness of the objects since higher pressure can be observed with thicker objects. Fused with the data from the capacitive and inductive sensor, resistive sensing could potentially improve the robustness and accuracy of object recognition. My fabric resistive sensor implementation involved creating a three-layer structure with a piece of pressure sensitive material (e.g., velostat) sandwiched between two layers of conductive fabric.

**Sensing Tagged Objects Using NFC:** Tagged objects are sensed using Near Field Communication (NFC), which is a technique commonly used in applications involving contactless payments or tagged detection [143]. The technique uses alternating electromagnetic fields for sensing and transmitting data. When a NFC tag is triggered by an
electromagnetic interrogation signal from a nearby antenna coil, it transmits its data to the sensor coil. In my implementation, I carefully designed the coil layout and its circuit to ensure that the sensor can not only detect tags, but can also function as an inductive sensor.

**Sensor Design**

To develop these four different sensor modalities in a single package, a naive approach would be to stack them four discrete sensors on top of each other. However, this would increase the thickness of the sensor and complexity of the fabrication process and interface circuitry. To overcome this challenge, I designed the sensor in a two-layer structure (Figure 4.14) with the top layer composed of a grid of fabric resistive sensors with conductive electrodes for both capacitive and resistive sensing and the bottom layer composed of a grid of embroidered coils for inductive and NFC sensing. I designed my sensor to cover the space of a 100 mm x 100 mm region, which is roughly the size of a jeans pocket.

**Inductive-NFC Sensing Layer Design:** For object recognition, inductive sensing usually requires the sensor coils to be arranged in a grid layout to detect the rough geometry of the contact area of an object. The grid arrangement also ensures the NFC sensor is effective across the full area. However, a tag may not be recognized when it is placed between two adjacent coils. I overcame this challenge by introducing a small overlap between two adjacent coils (Figure 4.14). Note that the overlap may impact the sensing resolution of the inductive sensor in the 2D space. My initial test suggests that with my coil design (see details below), a 5 mm overlap works best for balancing the coverage of NFC and the sensing resolution of the inductive sensor in the 2D space.

To maximize the sensitivity to objects of different materials and shapes, the size and shape of the coils for inductive sensing can be determined using the approach described in a previous work [85]. Once the design of the coils is determined, they are shared by the NFC and inductive sensing circuits. A multiplexer can be used to swap between the two circuits, as shown in the circuit schematic in Figure 4.15. The challenge, however, is that
in practice, multiplexers introduce parasitic impedance that degrades performance. As such, I had to minimize the use of them in my design. Since the LDC1614 chip used in my implementation (more details later) supports four channels, this allowed us to implement my sensor using four coils without the need of any extra multiplexer except the one used for switching between the two circuits. This is why my current implementation has a grid of $2 \times 2$ coils. To support higher-resolution sensing the number of coils can be increased in the future with better hardware. Note that another limitation of the current inductive sensing hardware is that operation becomes unreliable when the inductance of the coils is below 4 uH. As such, I designed my coils with an inductance of 4 uH (e.g., square shape, number of traces = 5). Figure 4.14 shows the detail of my coil design.

**Resistive-Capacitive Sensing Layer Design:** To enable capacitive sensing I re-purposed the conductive fabric of the resistive sensor as the electrode for capacitive sensing. This approach is similar to what is described in zPatch [179]. However, unlike the previous work, recognizing objects and touch gestures requires the sensor to be arranged in a grid layout. In my implementation, I arranged the resistive sensors in a $4 \times 4$ grid. Note that the problem with the grid arrangement is the conflict between the design for the capacitive and resistive sensor. For example, the row and column electrodes of the capacitive sensor need to be electrically separated while the row and column electrodes of the resistive sensor
need to be electrically connected. Thus in my implementation, I left the resistive sensors disconnected from each other. Each sensor was connected to the sensor board separately. This setup works well for a pocket.

**Sensor Implementation**

In this section, I detail the hardware implementation of the pocket sensor.

**Fabrication:** The coils on the inductive-NFC sensing layer was created by stitching conductive wires onto a cotton substrate, as shown in Figure 4.16A. Since my design requires some of coil traces to overlap with each other, I had to use insulated wires to avoid short circuits. In my implementation, I used the standard enamel coated copper wire, widely used in the smart fabric industry. Based on some initial testings with a JGVA embroidery sampling machine [180], I opted for the 34 AWG wire (161 um diameter) as it is both thin and strong enough to stand the fabrication process. The enameled wire was applied to the fabric substrate using a fixation top thread (Polyneon #40 weight) interlocked with a bottom thread (polyester #150 weight).

To create the electrodes for the resistive-capacitive layer, I first stitched a sheet of conductive fabric (EeonTexTM NW170-PI-20) onto a cotton substrate. The stitches followed the grid layout of 16 square-shaped electrodes. The conductive fabric faces the inner side of the sensor to allow a contact with the middle layer of a pressure-sensitive material. Note that I chose the conductive fabric that was made of a non-metallic material (e.g., conductive polymer) to avoid interfering the signals of the other sensors (e.g., inductive sensor). Following the stitches, I cut the conductive fabric outside the electrodes using a cutting machine (Cricut Air Explorer). Next, I stitched a connection line from the top right corner of each electrode to the corresponding position of the pins on the sensor board using the same enameled wire. To prevent the fabric from being bent easily along the gap between two adjacent electrodes, I added a sheet of felt to hold the electrodes on their positions.

The resistive-capacitive sensor was completed by sandwiching a grid of 16 square-
Tasca circuit consists of a processing module, a sensor module, and a multiplexer module. Due to the negative impact on inductive sensing from the multiplexers’ parasitic impedance when the coils are shared between inductive and NFC sensing, I had to minimize the multiplexers in my circuit design.

I shaped pressure-sensitive fabrics (velostat) between the two electrode layers using stitches (Figure 4.16B). Note that in order for the pressure sensor to work properly, contact between the velostat layer and the connection lines of the electrodes needs be avoided. As such, the top right corner of the velostat pieces was removed and replaced with a piece of felt to create an insulation between the top and bottom electrodes.

The entire sensor was completed by stitching the two individual layers together into one piece. I also added a fabric ground shield on the bottom of it. To develop the sensor in a pocket form factor, I stitched it onto a piece of denim with an opining on the top. I then replaced the original pocket of a pair of jeans with my pocket sensor by attaching the sensor to the jeans using velcro (Figure 4.16C). In my current implementation, I placed the sensor in the front-right pocket as it is easy for people to perform touch input.
Figure 4.16: Tasca sensor. (A) The inductive-NFC sensing layer. The coils were created by embroidering enameled copper wires onto a cotton substrate. (B) The resistive-capacitive layer. The sensor grid was created by stitching the velostat pieces between two conductive fabric layers. (C) Tasca is installed on a front pocket of a pair of jeans using velcro

**Customized Sensor Board:** My customized sensor board is comprised of a multiplexer module, a sensor module and a processing module (Figure 4.15 and 4.17).

The multiplexer module connects the electrodes and coils to the corresponding sensor circuit. My implementation has three four-channel 2:1 multiplexers (TMUX1574, Texas Instruments), two for the system to switch four coils between the inductive and NFC sensing circuits and the other one for the system to switch between the capacitive and resistive sensing circuits. Additionally, I used four two-channel 4:1 multiplexers (FSUSB74, On Semiconductor) to handle the 16 electrodes for the capacitive and resistive sensor. Further, I used an additional two-channel 4:1 multiplexer (FSUSB74, On Semiconductor) as the RF switching component to handle the 4 coils for NFC. No multiplexer is needed for the inductive sensor as the LDC1614 chip supports four channels.

Mounted on the multiplexer module, the sensor module hosts the necessary circuitry for the four different types of sensors. For inductive sensing, I used a LDC1614 4-channel inductive sensing chip from Texas Instruments running at a circuit capacitance value of 680 pF (3 MHz operating frequency). The capacitive sensing was implemented using self-capacitance with a FDC 2214 capacitor sensing chip also from Texas Instruments. The resistive sensing was implemented using a voltage divider circuit for monitoring the change in the resistance of the pressure sensor. Finally, the NFC circuit used a MFRC522 reader
chip. Note that the strength of RF signal fluctuates when small changes occur in coil inductance due to inevitable sensor deformation during use. This affected the signal strength of NFC sensing. To allow for the strength of the RF signals to remain at a relatively constant level, I included two programmable capacitors (NCD2100, IXYS) in the system. This enabled dynamic adjustment of signal strength based on the coil inductance detected using the inductive sensor.

Finally, mounted on top of the sensor module, the processing module is composed of a Teensy 3.6 development board. It reads the data from all the sensors at 10 Hz and transmits to a laptop via USB for computation.

**Pluggable Connection:** The sensor board was connected to the fabric sensor through a pluggable interface implemented using pin headers (Figure 4.17). I soldered an array of male headers at the end of the sensor’s connection lines and female ones on the sensor board. To ensure that the entire system can be worn on the body comfortably, I placed the rigid sensor board on the outside of the jeans.
Figure 4.18: The heatmap images of the raw sensor data showing the capacitance, pressure, and inductance footprint of a wallet with coins.

**Data Processing**

For every 100 ms, the sensor reports a $4 \times 4$ grid of capacitance values, a $4 \times 4$ grid of pressure values, a $2 \times 2$ grid of inductance values, and NFC data. All the data, except from NFC, was used for object or gesture recognition. Before the raw sensor data was used for recognition, it was smoothed using a median filter with a sliding window of size 10. I then subtracted background noise from the sensor values using a 2D noise profile, created by averaging the sensor readings at all the locations of the sensor with a sliding window of size 10. For every 5s, I also updated the noise profile if the deltas between current sensor values and the initial ones were classified as noise by a machine learning model. Upon the presence of an object or hand, I upscaled the sensor data to a $240 \times 240$ heatmap image using linear interpolation. Figure 4.18 demonstrates an example of a wallet with coins inside it and its corresponding sensor footprint shown in the heatmap image.

**Object Recognition**

My system recognizes objects based on the inductance, capacitance, and pressure footprint of the contact area of the objects. For the data collected from each type of sensor, I derived 33 material- or pressure-related features and 53 shape-related features. I also added 16 pressure data collected when the sensor was in the idle model (e.g., without the presence
of an object or hand). This data inferred how tight the sensor was worn on the user’s body. In total, 274 features (see Table 4.2) were collected and used to train my machine learning model. I used Random Forest in my implementation because it has been found to be accurate, robust, and computationally efficient in applications involving small wearables and interactive fabrics [85, 79].

| Shape-related Features (53) | • Local Binary Pattern (36)  
|                            | • Hu Moments (7)  
|                            | • Object Area (1): Number of pixels the object covers  
|                            | • Object Edge (1): Number of pixels on object’s edge  
|                            | • Average Distance (4): Average distance from object’s pixels to object’s center of gravity and geometric center (2), average distance from object’s edge pixels to object’s center of gravity and geometric center (2)  
|                            | • Object’s center of gravity and geometric center(4)  

| Material-related or Pressure-related Features (33) | • Statistical Functions (13): Sum(1), Mean(1), Max (1), Binned Entropy (1), Local Maximum Numbers (1), Median (1), Quantiles (3), Count above/below mean (2), Variance(1), Absolute energy of the object’s pixel values (1)  
|                                                    | • Ten-Fold Stats (20): Sort and divide the object’s pixel values into 10 folds and average for each fold (10), Divide grayscale values (e.g., 0 255) into ten intervals and count the number of the pixels in each interval (10)  

Table 4.2: The feature set extracted from each sensor data for training my machine learning model.
Finger Gesture Recognition

If a hand was recognized, the system switched to the gesture recognition mode. Finger gesture recognition assumed the palm remains in a relatively stable position. This allowed us to use background subtraction to remove the palm in the heatmap image. I then detected the moving fingers by using OpenCV’s blob detection to look for blobs smaller than a threshold size. Gestures were recognized if the finger’s moving distance exceeded a certain threshold. This is similar to the method described in [181]. The normal force of the finger pressing the sensor was detected using the resistive sensor.

4.4.4 Interaction Techniques

Tasca’s unique sensing capabilities enable five different input modalities in one fabric-based sensor: object recognition, touch gesture, pressure input, and activity tracking. In this section, I describe several example applications to illustrate potential uses of these modalities through eyes-free, private, always-available, and context-driven interactions.

Object Recognition

Tasca understands what objects a user carries in the pants pockets. This enables richer contextual interactions in wearable scenarios beyond what is currently offered by existing wearable devices, such as smartwatches or head-worn displays. For example, knowing what the user has or does not have in their pocket, Tasca can provide better personal assistance. In my implementation, the system can detect whether the user carries loose change (e.g., coins) (metallic) in their wallet that can be used to pay for street parking or to purchase an item from a vending machine. A reminder is sent to the user before they travel, if their empty wallet (non-metallic) is detected.

In VR games, tagged objects (NFC) can be used as tangible tokens to enable a more immersive gaming experience by allowing the user to physically interact with virtual items.
Figure 4.19: An NFC tagged toy sword is used as a tangible token in a VR game. (A) A user can carry the sword in their pocket, which is used as a physical extension of the user’s virtual storage for their weapons in the game. (B) The user can switch to the sword from the fist by grabbing the sword token from their pocket.

in the digital world. For example, when the user encounters a weapon in a game, they could pick it up and place its physical counterpart (e.g., the weapon token) in their pants pocket. This adds the item to the user’s virtual inventory. When the user wants to use it, they can grab the physical token from their pocket (Figure 4.19).

**Gestural Input Using The Hand**

Unlike existing work [159], my system can differentiate between the hand (body) and other objects. Since the pocket is where hands can naturally reside, Tasca provides a useful input mechanism for a user to interact with computing devices. For example, a user can use touch gestures inside the pocket to interact with a head-worn display or a smartwatch. This subtle and eyes-free input method can be useful especially in the public settings, where repeatedly interacting with the device might be considered inappropriate. When Tasca is used to interact with a smartwatch using the same-side hand wearing the smartwatch (Figure ??B), one-handed interaction becomes possible on a smartwatch. This type of interaction can be beneficial in situations where the other hand is occupied by holding objects or is busy with other tasks.
Pressure Input

In addition to 2D touch gestures, pressure as an input modality enables a new dimension of interaction with computing devices through the pocket. Prior research has shown the promise of pressure sensing in gestural input on smartphones [182] or text input on small head-worn displays [183]. In my implementation, a user can perform directional swipes with two levels of pressure, low and high. Unlike the pressure input on a rigid-body touch panel, where the amount of normal force is only perceived through the fingertip, pressing inside a pocket allows the user to feel the normal force also through the thigh. Therefore, in-pocket touch input through pressure not only expands the vocabulary of touch gestures, it also enriches the haptic feedback that the user can perceive to better support eyes-free input.

Activity Tracking

Sensing the hands of a user inside the pants pockets can also provide rich contextual information related to their current activity. For example, a hand inside a pocket while interacting with a smartphone using the other hand indicates one-handed use of the smartphone. As such, the smartphone UI can be adjusted accordingly to facilitate input using the thumb (e.g. a narrower keyboard for one-handed typing) (Figure 4.20).

In social scenarios, the body language expressed by the user putting their hands in the pants pocket is often associated with certain social meanings. For example, hands in a pocket when standing can be considered as a sign of low confidence [184]. Self-correction is often hard since body posture is driven by a subconscious process. The system can be setup to notify the user about their hand position through the vibration of a smartwatch.
Figure 4.20: My system can sense user activity by tracking if a user puts their hands in the pocket. (A) When typing using both hands, the keyboard is configured in its maximum width. (B) The system detects that the user puts their hand in the pocket. (C) The system automatically switch the keyboard to a narrower version to facilitate typing using one hand.

4.4.5 Evaluation 1 - Object Recognition

The goal of this study was to validate the object recognition accuracy of my prototype and its robustness against individual variance among different users.

Participants and Apparatus

Ten participants (age: 18-34, 4 males, 6 females) were recruited to participate in this study. The size of their hip ranged from 33 inch to 40 inch (average: 37.5, SD = 2.2) and the size of their upper thigh ranged from 17 inch to 22 inch (average: 20.0, SD = 1.6). I customized my sensor for each participant by installing it in the front right pocket of a pair of skinny jeans purchased at their size. They wore the jeans during the study. My study was conducted by following an approved Institutional Review Board (IRB) protocol developed specifically to ensure the safety of my researchers and participants during a pandemic.

Objects

I tested my prototype using 10 objects that are often carried by people in their pants pocket (Figure 4.21). The tested objects vary in geometrical (e.g., size and shape) and material properties. Some of them were pure metallic or non-metallic, while many were made of a
mixture of metallic and non-metallic materials. For the non-metallic objects, I included a leather wallet and a bottle of hand sanitizer. For the metallic object, I included a door key. For the hybrid ones, I included a signature pen, multitool knife, car key, earbuds charging case, and LG Rebel 4 smartphone. I also purposefully included an empty sanitizer bottle and a few coins in the wallet to measure how well my system can recognize different statuses of a container. For example, I tested how well the system can recognize if the sanitizer bottle is empty or if there are coins in the wallet. I let the participants to randomly choose how many coins they wanted to put in the wallet. I ended up collecting the data ranging from five to ten coins. Further, I included the hand in my study to test how well my system can differentiate between the hand and all the other tested objects.

**Data Collection**

The study was conducted with my participants performing the task in a standing position to simulate the common use scenarios of a pants pocket. Participants were asked to place each of the tested object in their pocket 10 times at a random order. Further, to test whether user activities such as walking may cause confusion to the system between the tested objects, I collected noise data by asking participants to walk for 30 seconds. In total, it took about 20 minutes for the participants to complete the task. In total, I collected 1100 samples (10 participants × 11 objects × 10 repetitions) for analysis.

**Result**

I present my results using within-user accuracy and cross-user accuracy. Additionally, I discuss how data from different sensors contributed to the accuracy of object recognition.

**Within-User Accuracy:** Within-user accuracy is the measurement of prediction accuracy, where training and testing data are from the same participant. For each participant, I conducted a five-fold cross validation, where 4/5 of the data was used for training and the remaining data used for testing. I then calculated the overall within-user accuracy by
Figure 4.21: (Left) Objects tested in the study: (A) LG Rebel 4 smartphone, (B) earbuds charging case, (C) key, (D) full bottle of hand sanitizer, (E) empty hand sanitizer bottle, (F) wallet without coins, (G) wallet with coins, (H) signature pen, (I) car key, (J) multitool knife. (Right) The within-user confusion matrix and cross-user confusion matrix.

averaging the results from all the participants. The result yielded an accuracy of 92.3% (SD = 3.2%). Figure 4.21 shows the confusion matrix. A close look at the study result reviewed that most objects received an accuracy over 90%. The noise could be reliably distinguished from the tested objects and hand. The major source of error, however, was from the confusion of the system between the full and empty bottle of hand sanitizer, which suggests that very small differences in the material property of the objects is still challenging to distinguish. One of the potential reasons of this issue is related to the small inconsistency in sensor readings at different locations of the current prototype. Further, the tightness inside the pocket also varied at different locations. When the variation in sensor readings at different locations was somewhat closed to the impact caused by the change in the material property of an object (e.g., with vs without liquid in this case), a reduction in system performance was observed. I were thus interested in knowing whether system performance could be improved after removing the data from the most confusing object. I found that the accuracy increased to 95.5% (SD = 2.2%) without the empty hand sanitizer. Aside from the hand sanitizer, the system also sometimes confused the multitool with the signature pen. This is primarily due to the similarity in the shape and material of these two objects. I except that this problem can be solved by improving the 2D resolution of the sensor.
**Cross-User Accuracy:** Across-user accuracy (or universality) measured how well my model works across different users. I calculated the accuracy by using the data from nine participants for training and the remaining one for testing. The overall accuracy was then calculated by averaging the accuracy of all the ten combinations of training and test data. The result yielded a 81.3% accuracy (SD: 6.0%). The reduced accuracy was expected as the sensor deformed differently across the users, which had made it more difficult for the machine learning model to handle. Figure 4.21 shows the confusion matrix. Similar to the within-user condition, the full (57%) and empty hand sanitizer bottle (53%) contributed to most of the classification errors. Additionally, objects that are similar in shape or made of similar materials began to be less distinguishable across different users. Examples include signature pen (87%) versus multitool knife (87%) and wallet (64%) versus empty hand sanitizer bottle (53%). The decline in system performance is primarily because of the individual difference in the shape and diameter of the thigh, which had led to variations in the footprint of the tested objects. For example, the footprint of the hand sanitizer was wider for the participants who had plump thighs than those who had slim thighs because the sensor was worn flatter in the former case, thus increasing the contact area between the objects and the sensor. Removing the signature pen, full and empty sanitizer bottle from the training/testing set increased the recognition accuracy to 90.3% (SD: 6.2%). This is in fact quite promising as it suggests that it is technically feasible to use a general model across different users without an impact on the type of objects that the system can correctly recognize.

**The Contribution of Different Sensors:** Aside from recognition accuracy, I were also interested in understanding how system performance was affected by the availability of the data from different types of sensors in my system. I included the analysis of the same set of objects except that the most confusing ones (e.g., empty hand sanitizer bottle in the within-user condition) were removed so the system worked in an "ideal" situation. I then calculated the recognition accuracy using only a subset of the sensor data. As shown in
Figure 4.22: The within-user and cross-user accuracy using a different sensor data set. The error bars are two standard deviations.

Figure 4.22, object recognition accuracy was affected by which sensor(s) was involved and how they were combined. As one may expect, no single sensor could reliably handle my diverse set of objects. However, if I combined any two of the sensors, the accuracy was improved. For instance, the combination of capacitive and resistive sensing yields a better accuracy (within-in user: 80.5%, SD = 6.7%; cross-user: 55.1%, SD = 9.4%) than using the capacitive (within-in user: 73.0%, SD = 5.7%; cross-user: 47.3%, SD = 6.2%) or resistive sensing alone (within-in user: 73.2, SD = 11.3; cross-user: 42.9%, SD = 6.6%). This suggested that even the pressure data alone was unreliable for object recognition, it worked well as an addition to improve the performance of capacitive sensing when the sensor was deformed by the body. The same pattern was observed for inductive sensing with performance increased to above 80% for both within- and cross-user conditions. Considering that the 2D sensing resolution for the inductive sensing is quite low in the current implementation (e.g., 2 × 2), I expect that the overall accuracy of my system can be improved by improving the 2D resolution of the inductive sensor.

4.4.6 Evaluation 2 - Gesture Recognition

The goal of this study was to measure how accurate my system can sense some of the most common touch gestures in daily computing tasks. I were also interested in understand-
Figure 4.23: Left: An illustration of the tested gestures. Right: A confusion matrix showing how well the system can detect the gestures.

The participants were asked to use my system to perform the gestures with different levels of pressure (e.g., low vs high).

**Participants and Apparatus**

I invited the same group of participants to participate in this study. The study apparatus was also same as in Evaluation 1.

**Gesture Sets**

I included in my study four directional strokes (left, right, up, down) that are commonly used on touchscreen devices or for navigating large workspaces (e.g., a map or long list) (Figure 4.23). Note that my sensor can detect other types of common gestures for interactive fabrics (e.g., deforming the fabric [44]) but according to my pilot study, most of them require relatively large hand motion that are uncomfortable to perform in a small jeans pocket. For each tested gesture, participants were asked to perform the gesture with low and high pressure. Note that my pilot study suggested that the perception of low versus strong force varied across different people. Therefore, I let each participant to perform the force gestures at their own pressure levels.
Data Collection

Similar to Experiment 1, the study was conducted with participants performing the task in a standing position. Before I started the experiment, participants were given several minutes to learn the 8 gestures. During this short training session, I also customized the pressure threshold for each participant. After this short training session, each participant performed a gesture inside the pocket using their right hand (Figure 4.23). The order of the tested gestures was randomly assigned. Each gesture was repeated 10 times and the entire experiment took less than 20 minutes to complete. In total, I collected 800 samples (10 participants × 4 gestures × 2 forces × 10 repetitions). Real-time recognition accuracy was recorded for analyzing the study results.

Result

Overall, my system yielded an average gesture recognition accuracy of 96.1% (SD = 3%). The confusion matrix shown in Figure 4.23 suggests that 6 out of the 8 tested gestures could be correctly recognized by the system with an accuracy of over 95%. Some of the gestures were harder to perform than the others. For example, users with long nails often performed the light swipe up (94%) by scratching the sensor using the nail. This has caused difficulties for the system to properly detect the touch gesture. Further, performing the swipe down gesture (90%) using two levels of pressures was more challenging than the other tested gestures because participants naturally exerted more force when pushing the finger downwards even in the low pressure condition. It was thus more challenging for the system to distinguish between the two swipe down gestures with different levels of pressure.
4.4.7 Evaluation 3 - NFC Tag Detection

The goal of this study was to evaluate the robustness of my NFC sensor in different tag position and distance to the sensor.

Participants and Apparatus

The same group of ten participants were invited to participate in this study. I tested my system using two common types of NFC tags, card and key tag (Figure 4.24). The distance between a tag and the sensor was controlled by attaching the tag to an acrylic sheet of certain thicknesses (Figure 4.24).

Data Collection

The data was collected with the tag placed in the center as well as at the four corners of the sensor. I chose these locations because sensor signals are weaker at the edges of the coils. Similar to Experiment 1 and 2, participants were instructed to place the tags in the tested locations in a standing position. Note that the tags may appear at a small distance away from the sensor when attaching to an object of a certain thickness. So in this study, I also included three tag distances at each of the tested locations (near, medium, and far). In the near condition, the tag was placed in a direct contact with the sensor. The medium and far conditions were controlled at 3 mm and 6 mm (about the thickness of a smartphone) respectively for the key tag. Note that sensor signals are stronger for the card, so I increased the distances for medium and far conditions for the card to 10 mm and 20 mm, which is about the thickness of a wallet. Each trial was repeated 3 times and the entire study took less than 15 minutes to complete. In total, I collected 900 samples (10 participants × 2 tags × 5 locations × 3 distances × 3 repetitions) for analysis.
Figure 4.24: Recognition success rate of my system with the NFC tags placed at different locations and distances to the sensor

**Result**

Figure 4.24 shows the result of the experiment. Overall, the recognition success rate for the card was 98% across all the locations and distances. The success rate of the card dropped at both corners near inner thigh (e.g., left corners) at the longest distance of 20 mm. I found that the signal was weaker on that side of the pocket, especially for those who had slim thighs. This is possibly because of the increase in sensor deformation at the far end, which consequently increased the distance of the tag. The system was able correctly recognize the small key tag at all the tested locations in the 0 mm and 3 mm distance conditions. However, due to the drop of the signal strength at the 6 mm distance, the key tag could not be reliably detected.

### 4.4.8 Limitations and Future Work

In this section, I discuss limitations of my work and suggest future research for exploring the space of interactive pockets.
Sensor Hardware

I demonstrated that my system can recognize some everyday objects, as well as simple finger gestures. The sensing accuracy of my system for both object and gesture sensing can be improved in the future with the availability of better hardware (e.g., multiplexer). As shown in my study, the number of electrodes and coils impacted recognition accuracy of the tested objects in both within- and cross-user conditions. I expect that including a denser array of electrodes and coils will allow the system to be more capable of sensing user input. Furthermore, going beyond the focus of my current research, incorporating different types of hardware design for sensing will likely open opportunities for a completely new set of applications in pocket-based computing. As a next step for future research, I will focus on new capabilities, such as wireless power transfer for charging electronic devices inside the pocket.

Effect of Body Motion

As a flexible wearable sensor, the readings of my prototype are affected by the motion or posture of a user’s body. For example, actions such as sitting down or jogging may introduce noises that can be hard to handle using a machine learning model trained in a stationary condition. My initial investigations demonstrated that noises caused by walking had no noticeable effect on how well the system could distinguish between different types of objects or hands. My next step is to investigate whether my system can handle other types of body motion, identify other issues unique to the context of an interactive pocket, and explore practical solutions to overcome the challenges.

Beyond Jeans Pockets

In the context of object recognition, my current research focuses on pockets in a pair of jeans. This led to the design of my sensor primarily using contact-based techniques that require an object to be in contact with the sensor. This requirement can mostly be guaranteed
on a pair of jeans, especially with popular skinny styles as the pockets are tight to the body. The shape of the sensor conforms with the user’s body, thus largely reducing the possibility of it to be deformed in a random way and consequently affect sensor readings. However, in other wearable scenarios beyond jeans (e.g., pockets in a hoodie, or a jacket), firm contact between an object and the sensor may not be guaranteed, so it is unknown how well my current object detection method can work in these situations. Future work can further look at understanding the challenges for my system to be used in scenarios beyond jeans and identify novel contactless solutions (e.g., EM- or thermal-based sensing) to overcome the challenges.

Fabric Flexibility

With multiple layers of electrodes, coils, and substrates, it is expected that my implemented jeans pocket is harder than it was before instrumentation. Preserving the softness of the fabric sensor is also an important consideration in my future explorations. I see it as an interesting future research direction to investigate ways to optimize the sensor based on the material properties of the substrates and wires, which can lead to improvements in softness and comfort of the sensor.

4.4.9 Conclusion

In this paper, I demonstrate a technique that combines inductive sensing, capacitive sensing, resistive sensing, and NFC into a multi-layered fabric that is integrated into a pocket of a pair of jeans. First, I discussed my sensing principle and approaches to optimize the layering, which was selected based on careful research. With a ten-participant study, I found my approach demonstrated a 92.3% with-user classification accuracy with 11 different objects, 96.4% accuracy for gesture recognition and 100% accuracy for NFC tag detection at a maximum distance of 3mm (key tag) or 10mm (card). I demonstrated a set of new application scenarios, and believe it is an important step to enabling new types of interac-
tions with an everyday part of garments, as well as expanding the broader input space for interactive fabrics.
Chapter 5

Makeable Computational Materials

The ability of computational materials to sense user inputs is fundamental for the creation of smart everyday objects and environments, but another equally vital concept is makeability. Makeability refers to the ability of computational materials to withstand fabrication operations that are involved in creating everyday objects. For example, the makability of computational wood is the ability to withstand woodworking operations such as sawing, cutting, screwing and nailing. Computational materials with "good" makeability should allow the materials to be processed using the same tools and machines designed for their non-computational counterparts without any modification. It is a crucial concept because it allows the smart physical world to be created using the established methods, by which, the current physical world is created.

Furthermore, from the human perspective, crafting or making is a kind of human natural interactions with materials [185]. When humans perceive materials, they are naturally inclined to use them to make something. For example, when you perceive a piece of wood, you know you can cut and assemble it to make tools, artifacts and furniture such as wooden spoons, cutting boards, tables. This intuitive interaction is a form of material’s affordance that enables designers to use materials creatively for composing a variety of everyday objects. Same as regular materials, computational materials should preserve such affordance.
It empowers non-technical individuals to utilize computational materials to create "smart" everyday objects in a way that is naturally afforded to them.

Therefore, Part II of my thesis focuses on the makeability of computational materials. In a broad sense, makeability covers the interfaces, computing units and all the other electronics in computational material, but in this dissertation, I focus on the interface of computational materials, specifically in sensing part. In this dissertation, I exemplify how to make a interface of computational materials makeable through a research project called iWood, makeable computational plywood.

In this part, I worked on computational plywood because (1) like textile, plywood is commonly used to create daily objects and environments, including artifacts, furniture, and house infrastructure, which will largely complement my research on textiles; and (2) the challenges in makability are significant on computational wood (e.g., shorted circuits due to metal connectors) and also common on textile and other materials (e.g., paper), which means that my solution developed on plywood could be generalizable among other types of materials.

Lastly, in this part of my thesis, I choose to develop the contextual sensing using vibration based on triboelectric effect because (1) it complements the sensing technique I proposed in the first goal based on material properties; and (2) the triboelectric vibration sensor (more details later) has a common structure shared by in other types of sensors such as paper-like microphones [?]. Thus, the makeability problems that I address are also common among the other types of sensors. This, again, will largely make my research outcomes generalizable.

### 5.1 Background

My research in makeability was inspired by the line of work investigating cuttable sensors for rapid prototyping. An early example of cuttable sensors is Wimmer et al. [136] and
Holman et al. [64] ’s touch sensing strip that can be cut into different lengths to satisfy the needs of different applications. Similarly, Dementyev et al.’s sensor tape can also be cut into different lengths and is capable of sensing the deformation of the sensor and measuring the proximity of nearby objects [186]. Beyond cutting in a one-dimensional space, Olberding et al.’s multi-touch sensor sheet can be cut out in 2D shapes [50]. To allow the sensor to be robust against different types of damages caused by cutting, the authors introduced a new electrode layout based on the insights from physical routing topologies. Their work was validated in a study with six cutout shapes. Their result showed that the new design of electrode layout could allow 80% of the electrodes to remain functional after the sensor was cut into different shapes. Built on top of this work, Takahashi et al. [187] developed a cuttable coil grid for wireless power transfer using a method based on H-tree. The work in cuttable sensors inspired us in many different ways in the development of solutions to address the makeability problems in an interactive plywood material.
5.2 iWood

5.2.1 Introduction

The vision of ubiquitous computing heralds the future of smart home and work environments that can better understand and fulfill people’s needs [1]. However, this vision is still far from reality as most things in today’s world are not computing-powered, such as furniture made of wood or garments made of fabric. To allow computation and interactivity to better blend into everyday contexts, researchers have investigated ways to imbue them into everyday materials, of which, daily objects are made [4]. This way, the world made of computational materials becomes interactive automatically while still being able to preserve the look and feel of its non-computational counterparts. Innovations like interactive paper [49, 50, 51, 48, 52] and fabric [42, 43, 44, 45] all exemplify such efforts.

![Image Description](image)

Figure 5.1: (a) iWood is interactive plywood that can be used as a vibration sensor to detect user input and activities. What’s unique about iWood is it remains functional against common woodworking operations such as screwing and sawing. (b) Smart cutting board, table, and nightstand made of iWood that can sense user input and activities. (c) The cutting board can sense kitchen activities such as chopping and slicing, (d) The table can detect work activities such as writing, erasing, and stapling. (e) The nightstand can detect a user pumping lotion.
In this project, I present a new type of computational materials created based on plywood, a type of wood commonly used in furniture, artifacts, floors, and building infrastructures. Beyond what plywood offers as a material, my prototype (called iWood) detects the subtle vibrations caused by users’ gestures and activities based on the triboelectric effect [188]. iWood is makeable, meaning that it can be sawed, nailed, or screwed together to make smart objects. Things created using iWood inherit the material’s sensing capability and can detect a variety of user input and activities based on their unique vibration patterns (10.1.b - 10.1.e). For example, a smart desk made of iWood can detect and log the user’s work activities, such as writing and erasing, enabling new applications for skill development or personal reflection.

Like regular plywood, iWood is low-cost. I implemented my prototype using a layer of triboelectric material sandwiched between two layers of electrodes, each attached to a plywood substrate (Figure 5.1.a). The sensor has the same structure as those commonly used in triboelectric nanogenerators (TENG) [52]. As a low-cost alternative to other approaches like piezo, it is much easier to implement on interactive plywood. I tested the makeability of iWood with common woodworking operations, such as sawing, drilling, screwing, and nailing identified from 300 woodworking projects and tutorials documented in books, DIY websites, and video sharing platforms. My result showed that the structure of the sensor, composed of overlapped electrodes (also common in other types of sensors, such as pressure sensors) is susceptible to fabrication operations due to short circuits caused by metallic connectors like screws and nails passing through the electrodes. To address this problem, I replaced the single electrode with multiple ones arranged in a grid layout. The top and bottom electrodes stagger with each other to cover a separate part of the sensing area, thus significantly reducing short-circuiting. Through a series of experiments and machine simulations, I carefully chose the size of the electrodes, the type of triboelectric materials, and the bonding method of the sensor layers to optimize the sensitivity and fabrication complexity of iWood. In a controlled experiment, I measured the sensing performance of the
table, nightstand, and cutting board over four input gestures and 12 common work and kitchen activities (e.g., writing on a table or tenderizing meat on a cutting board). My results suggested that in many tested conditions the smart items could achieve a recognition accuracy of over 90%.

The contributions of this project include: (1) an interactive plywood prototype that can be used to create smart wooden objects to sense user input and activities using vibration sensing; (2) the notion of makeability as an important consideration in the development of interactive materials; (3) insights and solutions to address iWood’s makeability issue; (4) the result of an experiment measuring the performance of iWood; and (5) usage scenarios demonstrating unique applications enabled by iWood as a makable interactive material.

5.2.2 Related Work

In this section, I discuss the related research about interactive wooden artifacts and vibration sensing based on the triboelectric effect. I do not discuss inputs on interactive materials and cuttable sensor because they were discussed in the previous chapters.

Interactive Wooden Artifacts

Similar to fabric and paper, wood as one of the most common materials has also received attention in the crafting community. However, within the existing research, most have primarily focused on attaching sensing devices to off-the-shelf furniture or artifacts made of wood [22, 189, 190, 191, 192, 193, 194] rather than developing wood into an interactive material. Examples of this body of research include tables, walls, and floors instrumented with vibration sensors to detect touch or gesture events [189, 22, 193], user activities (e.g., cutting, typing, walking) [192], and the presence of some of the daily items on a table (e.g., glass, phone, coil, paper cup) [194]. Although it is an effective way to bring interactivity to the exiting environment, the attached sensors and devices often do not blend well into the traditional aesthetics of wooden objects. While some of the work has its sensing unity well
hidden inside walls or floors [192], the sensor has to be customized for every new object because they were developed as an ad-hoc addition to the objects rather than a material.

The difference between iWood and the existing interactive wooden artifacts is that iWood was developed as an interactive material. Its sensing capability is an important property that must be inheritable by anything that is made of iWood, despite how the material is processed when an object is made.

**Vibration Sensing based on the Triboelectric Effect**

My sensing technique is based on the triboelectric effect and particularly triboelectric nanogenerator (TENG), which is a technology developed based on the principle of triboelectrification and electrostatic induction. It converts mechanical energy to a correlated electrical response and has been widely used in energy harvesting applications due to its high efficiency for energy transfer at low frequencies [195]. For example, Hao el al. [196] proposed a wood-based triboelectric nanogenerator (W-TENG) to produce electric output to power microelectronic devices. Sun et al. [197] improved W-TENG to harvest more energy by using a triboelectric material made of chemically functionalizing wood. Methods based on the triboelectric effect have also been used for sensing in a wide variety of applications to detect mechanical motion such as pressure [198, 199], vibrations [200, 201, 52], speed of wind [202], rotation of a disk [203], and acceleration of an object [204]. As mentioned earlier, this technology was also used as a sensing mechanism to instrument floors and tables to detect user activities, such as falling [205].

Sensors based on the triboelectric effect can be implemented in several different ways. For example, it can be implemented using two layers of triboelectric materials (one positive and one negative), each connected to a layer of electrode. The triboelectric materials are used for triboelectrification and the electrodes are used for harvesting triboelectric energy. An alternative approach, which can enable a simpler structure, uses one of the electrodes to also serve as a positive triboelectric layer. The nature of triboelectrification allows it
to be implemented using low-cost and lightweight materials such as polytetrafluoroethylene (PTFE) film (triboelectric layer) and copper films (electrode layer). In comparison to other types of vibration sensors, such as piezoelectric ceramic or PVDF poled piezoelectric film, techniques based on triboelectrification are cheaper and easier to implement [52], thus making them suitable for plywood, especially for development at scale.

In comparison to the existing implementation of the sensing techniques based on the triboelectric effect, I opted for a different electrode design to address the makeability issues associated with common woodworking operations. My approach is also unique in that I studied how the implementation options like the size of the electrodes, the type of triboelectric materials, and the bonding method of the sensor layers may affect the sensing performance and fabrication complexity of iWood.

5.2.3 Sensing Principle

iWood’s sensing technique is based on the triboelectric effect caused by the contact and separation of a positive and negative triboelectric material (layer) [204]. When the two layers are pushed to contact with each other, the negative triboelectric layer gains electrons from the positive triboelectric layer, becoming negatively charged. The positive triboelectric layer becomes positively charged. When the subsequent propagation of vibrations separates the two triboelectric layers, a potential difference is induced between the electrodes connected to them. This causes the current to flow from the electrode of the positive triboelectric layer to the electrode of the negative triboelectric layer (with the presence of an external load between the two electrodes). When the two triboelectric layers are pushed towards each other again, a current flow in the reversed direction occurs, which completes the cycle of electricity generation. Even the contact and separation take place at a micro scale, the voltage signals generated by the triboelectric effect can still be measured for sensing purposes.

The sensitivity of the sensor or the voltage signals generated between the electrodes in
an open circuit can be described as the following equation in theory [204]:

\[ V_{oc} = \frac{Q}{C} \]

where \( Q \) is the density of the electric charge on the surface of the triboelectric layers and \( C \) is the equivalent capacitance between the electrodes. The surface charge density is determined by the material choice of the triboelectric layers [206]. Materials with a large surface charge density are preferred. For the negative triboelectric layer, polytetrafluoroethylene (PTFE) film is widely used [45]. For the positive triboelectric layer, there is a wider range of options with the most common ones including wood, copper, nylon, and polyurethane (PU) [206]. The capacitance is determined by the maximum distance between the triboelectric layers when they are pushed away from each other due to vibration. The wider the triboelectric layers are separated, the smaller \( C \) is, which leads to larger \( V_{oc} \) and thus stronger signals. In my work, the maximum distance is determined by how tightly the triboelectric layers are bonded to each other.

### 5.2.4 iWood V1: Interactive Plywood

We developed an initial prototype to investigate the technical feasibility and the makeability of my approach. My prototype consisted of a 0.2mm thin polytetrafluoroethylene (PTFE) film, sandwiched between two layers of 0.1mm thin copper film, which were then bonded to a substrate of plywood board (100mm \( \times \) 100mm \( \times \) 2.54mm) on each side (Figure 5.2). In this implementation, one of the copper films not only serves as an electrode but also acts as the positive triboelectric layer [204, 206]. This copper film was loosely attached to the PTFE film along the four edges to allow space between them for bouncing. Note that the other copper film, which served only as an electrode, was glued completed to the PTFE film. This was required to limit the occurrence of triboelectrification only on the opposite copper film (the one serving as the electrode and positive triboelectric layer) to maximize
signal strength. Alternatively, a sensor structure composed of two triboelectric layers and two electrode layers could be used [204]. I did not choose this approach to allow the sensor to remain as simple and low-cost as possible, making it easier to produce at scale. Since this prototype was primarily developed for investigating makeability issues, I did not optimize the sensor, the choice of triboelectric materials, and the bonding method.

Figure 5.2: iWood v1 is made of a 0.2mm thin polytetrafluoroethylene (PTFE) film, sandwiched between two layers of 0.1mm thin copper film, which were then bonded to a substrate of plywood board on each side.

5.2.5 Study 1: Makeability of iWood V1

In this section, I present my approach to measuring the makeability of iWood V1.

Understanding Common Woodworking Operations

First, I sought to understand common woodworking operations through projects and tutorials documented in woodworking books, DIY websites, and video sharing platforms.

Method: We collected 300 woodworking projects for analysis, among which, 46 came from some of the bestselling wood-crafting books (e.g., The Handbuilt Home and Good Clean Fun). The remaining 254 were from Instructable, DoItYourSelf and Youtube websites. The projects and tutorials coming from the web were obtained using the keywords “wood-crafting” and “wood artifacts”, with search results sorted by popularity (picked randomly if this feature was not available). Among the 300 projects used in this study, 82 of them described the fabrication of small artifacts such as cutting boards or coasters, 185 were related to furniture like chairs and desks, and 33 were related to houses and buildings.
To categorize the woodworking operations used in these projects, I used an open coding process.

**Result:** In total, I found 14 common woodworking operations, which can be categorized into 9 categories: sawing/cutting/laser-cutting, drilling, screwing/nailing, gluing, sanding, painting/staining/oiling, bending, soaking, and carving. Below, I briefly explain each category.

*Sawing/cutting/laser-cutting (O1):* This is a category, which includes operations to quickly cut plywood into parts of different shapes and sizes. Sawing is an operation that uses a disc or blade with teeth on its edge to cut wood. Cutting, in contrast, involves sharp tools such as a knife or scissors. In addition to sawing and cutting, laser-cutting is also widely used to cut plywood into a complex shape.

*Screwing/nailing (O2):* Screwing is an operation that rotates a short, slender, sharp-pointed metal pin around with helical thread to pierce pieces of wood to hold them tightly in place. Similar to screwing, nailing is an operation that joins pieces of wood together by inserting a small metal spike into the wood using tools such as a hammer.

*Drilling (O3):* Drilling is an operation that uses tools like a chisel, power drill, or reciprocating hammer to create a hole inside a plywood board.

*Gluing (O4):* Gluing is an operation that applies an adhesive substance on the wood surface to stick pieces of wood together.

*Sanding (O5):* Sanding refers to an operation that smooths or polishes a wood surface with sandpaper or a mechanical sander.

*Painting/staining/oiling (O6):* Painting, staining, and oiling are operations that spread a substance over the wood surface to either change or enhance the color of the wood, or coat the wood with a protective layer to prevent it from absorbing water or staining.

*Bending (O7):* Bending is an operation that forces the wood to be curved using clamps.

*Soaking (O8):* Soaking is a preprocess operation that wets the wood thoroughly to facilitate operations like bending to change the physical form factor of the wood.
Operation | Sawing//Cutting//Lasercutting | Drilling | Screwing//Nailing | Gluing | Sanding | Painting//Staining//Oiling// | Others | Occurrence %
---|---|---|---|---|---|---|---|---
98% | 85% | 79% | 72% | 58% | 67% | 5.5%

Table 5.1: The common woodworking operations summarized from 300 DIY projects.

**Carving (O9):** Carving involves cutting operations that only occur on the surface of the wood to create an aesthetic pattern or inscription.

Among these operations, six of them appeared in over 50% of the surveyed woodworking projects (Table 5.1), including Sawing//Cutting//Laser-Cutting, Drilling, Screwing//Nailing, Gluing, Sanding, and Painting/Staining/Oiling. Among these six operations, Sawing//Cutting//Laser-cutting, Drilling, and Screwing//Nailing cause serious damage to the sensor inside the plywood, so I consider them essential to study in this research.

### 5.2.6 Makeability of iWood v1

In this study, I sought to understand how these six woodworking operations may affect the sensor and its functionality.

![Figure 5.3: The woodworking operations that were tested on iWood v1 included (a) sawing, (b) drilling, (c) screwing, and (d) nailing](image)

**Method**

We developed several plywood sensors and performed sawing, drilling, screwing, and nailing on them (Figure 5.3). For each operation, I ensured that damage was caused to both PTFE and copper films. For sawing, I used a jigsaw to make cuts of random lengths in random locations. I then polished the cutting edge using sandpaper, which is often considered a necessary process after sawing. For drilling, I created holes in random locations using a
drill driver. Screws were driven in random locations using a screwdriver. Nails were also hammered in random locations. Each operation was repeated ten times. After each trial, I used an oscilloscope to check if the sensor was able to output vibration signals caused by the hand tapping the sensor.

**Result**

We found that only drilling did not cause any problems. All the tested sensors remained functional after drilling. In contrast, screws and nails caused the most critical problems as 100% of the trials involving screwing and nailing caused the sensor to malfunction (i.e., no signal reading). As a metallic object, when a screw or nail passed through the sensor, the copper electrodes became electrically connected (short-circuited), which led to no current flow within the circuit as the voltage potential equalized between the electrodes. It was even a more severe problem with the screw because when the screw was spinning inside the plywood and sensor, the electrodes were dragged and pushed to be in contact with each other and, unlike the nails, the short circuits remained even after the screw was removed from the sensor. A similar issue was observed in the sawing condition but less often (10% sensor failure). I found that the electrodes could be smashed to each other by the teeth of a saw. This issue, however, was gone after sanding because the thin copper films on the cutting edge were removed. Considering that sanding is often needed after sawing, I see sawing as less of a makeability problem. Based on my experiment result, I summarize that screwing and nailing are the main causes of short-circuiting, which is the most significant makeability problem of iWood in its initial implementation. Note that there are special types of nails that can be purchased with a non-conductive coating. It helps mitigate the problem, but non-conductive nails are not widely used in practice so a better solution is needed. Next, I describe my approach to addressing the makeability problem.
5.2.7 Redesigning iWood

My approach to makeability is inheritance - like many other material properties, the sensing capability of the interactive plywood should be inheritable by anything that is made of this material. To achieve this goal, I redesigned the sensor electrodes.

**Redesigning Electrode Layout**

The basic principle of my approach was to minimize the overlap between the electrodes. For example, instead of covering the entire PTFE film, the electrodes can be arranged to cover only half of it (Figure 10.4 10.5a). In the simplest way, the top and bottom electrodes can cover the right and left half of the sensor respectively. With this arrangement, the electrodes will not be easily connected by a screw. Note that, short circuits may still happen if a screw or nail appears on the shared edge of the electrodes. The problem, however, can be avoided by separating the electrodes with a gap wider than the diameter of the screw (Figure 5.4c Figure 5.6a). With this design, half of the PTFE film is in contact with the wood substrate, which now also serves as the positive triboelectric layer along with the copper film on the same side.

![Figure 5.4: Side views of the iWood structure. (a) In iWood v1, the screw inside the sensor connects the electrodes, thus causing a short circuit. (b) The new electrode design in iWood v2 removes the overlap between the electrodes to avoid short-circuiting. (c) A horizontal gap placed between the top and bottom electrodes further avoids short-circuiting along the shared edge of the electrodes.](image)

The issue with this simple modification is that the coverage of the two electrodes could be largely uneven after the plywood is cut into parts of different shapes. A sensor with uneven electrode coverage is susceptible to environmental electromagnetic (EM) noises, thus impacting the signal-to-noise ratio (SNR). Such noise could hurt the accuracy and robustness of iWood as an activity sensor. In an ideal situation with equal electrode coverage, EM
noises cause similar voltage signals on each electrode, thus can be almost canceled from each other in the output data, resulting in the minimum impact on the SNR. However, such a balance cannot be guaranteed inside a cutout (Figure 5.5a). In an extreme case, where one of the electrodes is mostly cut off from the sensor, the EM noises received from the opposite electrode will be largely included in the output signal, thus significantly degrading SNR. According to my test conducted under frequencies from 20Hz to 500Hz (frequency range of common user activities), the noise of a 610mm x 610mm sensor with only one electrode could be 100 times larger than the same sensor with well-balanced electrode coverage (Figure 5.5b).

Figure 5.5: Top view of the iWood electrode layout. (a) The top (red) and bottom (blue) electrodes, each covering half of the sensing area. (b) EM noises of an “injured” sensor, missing a big part of the blue electrodes. (c) iWood v2 opts for a grid electrode layout to avoid the extremely unbalanced coverage of the electrodes in (a).

To mitigate this problem, I opted for a different electrode design. Instead of using a single piece of the electrode on each side of the PTFE film, I opted for a grid layout with diamond-shaped electrodes connected in rows and columns (Figure 5.5c). In my design, the top and bottom electrodes stagger with each other with a 12.7mm gap to avoid short-circuits along the shared edges. I chose 12.7mm because it is the diameter of the largest wood screw that I found on the market. This new design allows the short-circuit problem to be largely avoided without sacrificing SNR (or sensitivity). Each electrode in the grid is connected to its four neighbors through connection lines of 2mm wide. This largely preserves the disconnection of edge electrodes in a cutout of any convex shape. In a concave shape like a star, some of the edge electrodes could be cut off from the sensor but most others will still
be functional. This helps avoid the extreme unbalance situation discussed above (as shown later in my study).

Note that the connections lines from nearby electrodes on the opposite side may still overlap with each other at intersections (e.g., where the red and blue lines cross each other in Figure 5c). Screws or nails in these locations may cause short circuits. The amount of overlap is determined by the size of the electrodes (Figure 5.6). For example, layouts with larger electrodes have fewer overlaps because there are fewer connection lines and electrodes. The trade-off, however, is in the lack of ability to tolerate the unbalance issue of electrode coverage upon cutting. Therefore, layouts with larger electrodes could be more susceptible to EM noises (Figure 5.6b). In contrast, layouts with smaller electrodes are less problematic upon cutting but can introduce more overlaps and short-circuiting issues (Figure 6c), which is a more serious issue. Next, I describe my approach to identifying an optimal size for the electrodes in a grid layout.

![Figure 5.6: (a) An illustration of what I mean by electrode size and the gap between the electrodes. (b) A square cutout containing large electrodes has the least amount of overlap between the top and bottom connection lines (more makeable) and the most unbalanced coverage between the electrodes (more susceptible to EM noises). (c) The same square cutout containing smaller electrodes has more overlaps between the connection lines (less makeable) but less of the unbalancing issue (better SNR).](image)

5.2.8 Simulation Study

An optimal electrode size can be found by testing size options with cutouts of different shapes, sizes, orientations, and locations inside a sensor. Each of these parameters has
many variations, leading to numerous combinations, which makes it infeasible for the study to be conducted manually. As such, I developed a software to simulate all different situations.

**Software Simulator**

My software simulated cutouts in different shapes, sizes, and orientations inside a virtual sensing area of 6100mm × 6100mm wide with electrodes of different sizes and a 12.7mm gap between the electrodes. The size of the sensing area is adjustable and the one used in my study was based on the size of a plywood board commonly found in DIY and crafting stores in my region. I chose six basic shapes to cover a range of simple and complex cutouts that are commonly seen in woodworking projects. These include triangle, rectangle, ellipse, star, carve-out, and hole-out (Figure 5.7). I varied the size of the cutouts by scaling them in the x- and y-directions six times by a factor of 1 to 6. This resulted in 36 different sizes and 30 variations of the basic shapes. The smallest cutout has a bounding box of 100 × 100mm, which is roughly the size of some of the small wooden artifacts found in my makeability study. I rotated each cutout from 0 to 180 with a step size of 22.5. Each combination of shape × x-scale × y-scale × rotation angle was then tested at different locations inside the virtual sensor along the x- and y-axis with a step size of 5mm in each direction, except when a cutout did not fit inside the sensor (e.g., at corners). Lastly, I varied the electrode size to cover a wide range of possibilities from 10mm to 120mm wide with a step size of 10mm.

![Figure 5.7: The six basic cutout shapes that were tested in the simulation.](image)

In each tested condition and for each electrode size, I calculated two scores with one indicating the probability of short-circuiting and the other one indicating the ratio between
the top and bottom electrode coverage. As discussed earlier, larger electrodes tend to have fewer short-circuiting problems but smaller electrodes tend to have better SNR. The Short-Circuiting Probability was calculated as the area of overlaps and any region that can be bridged by a screw of 12.7mm in diameter, divided by the area of a cutout. In my calculation, the connection lines were set as 2mm wide to be consistent with my implementation. The Balance Ratio was calculated as the smaller of the top and bottom electrode coverage, divided by the larger of them. When the coverage was calculated, I excluded the edge electrodes that were cut disconnected from the remainders. The Balance Ratio measures how well the coverages of the top and bottom connected electrodes are balanced. The higher this score is the better the coverages are balanced. In extreme cases, where electrodes on one side of a cutout are largely cut disconnected, the Balance Ratio will be close to zero. Thus, the Balance Ratio can also be used to detect the occurrence of extremely unbalanced coverage.

Result

We calculated the average Short-Circuiting Probability and aggregated them by electrode size, basic cutout shape, and cutout size in six groups: $100cm^2$, $(100, 400cm^2]$, $(400, 900cm^2]$, $(900,1600cm^2]$, $(1600,2500cm^2]$, and $(2500,3600cm^2])$. Figure 5.8 shows the Short-Circuiting Probability by electrode size, cutout shape, and cutout size group. Cutout shape and size had no observable impact on the likelihood of short-circuiting. As the size of the electrode increased, the chance for the electrodes and connection lines to be short-circuited decreased. With electrodes bigger than 80mm, the probability of short-circuiting became lower than 5%, which is very promising.

Figure 5.9 shows the Balance Ratio for the electrodes larger than 80mm. As expected, the Balance Ratio decreased with the increase of the size of the electrodes. The data also shows that the balance ratios of the electrodes bigger than 80mm are far greater than zero, meaning that extremely unbalanced coverage is rare. Among the different cutout shapes,
the star received the lowest scores. As expected, edge electrodes, especially the big ones, are more likely to be cut disconnected in a concave shape. When it happens to either electrode layer, the electrode coverage becomes largely unbalanced. The problem becomes more severe when the cutout is small (Figure 5.9b). For most other cutout shapes and sizes, the electrode coverages are relatively well balanced. Since my goal was to prioritize makeability while balancing signal clarity, I chose to use 80mm in my implementation.

5.2.9 Optimizing Implementation

Aside from the design of the electrode layer, the performance of iWood also depends on how the triboelectric and electrode layers are put together as a package. As discussed in
Section 3, the sensitivity of iWood is mainly affected by the bonding method of the sensor layers and the material choice of the triboelectric layers. Since in my implementation, one of the electrodes also served as the positive triboelectric layer, I were also interested in learning whether the material, of which, the electrodes were made would affect sensitivity. I performed three tests to answer these questions.

**Experiment Setup**

My study apparatus was implemented on a small piece of plywood board measured 100mm × 100mm wide and 5.08mm thick. Since makeability is not the focus of these studies, I only included a single electrode (100mm × 50mm) covering half of the PTFE film on each side, for the sake of simplicity. The rest of the implementation was the same as my prototype described in iWood v1. The electrodes were initially created using a copper film, which was later replaced by different materials along with other sensor modifications and bonding strategies based on the requirement of my tests. I created three replicated copies for each modified version to reduce the impact of fabrication variations on results.

The tests were performed in a controlled environment with the plywood sensor placed on a table. Vibration signals were generated using a vibration exciter, fixed in the center of the plywood using a high bonding double side tape. I controlled the vibration exciter using a wave generator to generate sine waves of 5V, with frequencies ranging from 20Hz to 500Hz at a step size of 4.8Hz. I chose this range because it covers the vibration frequencies caused by a wide variety of user activities in daily life [192]. The voltage signal at each tested frequency was acquired using an oscilloscope. EM noises were also collected for each test condition and sensor replicas to calculate SNR. The SNR data shown in the test results are the average of all the SNRs across the tested frequencies and conditions.
**Bonding Strategy**

In this test, I sought to understand how the bonding method of the sensor may affect sensitivity. A preferred bonding strategy should allow enough space for the PTFE film to bounce away from the copper film (one that also serves as the positive triboelectric layer), which means that they cannot be glued completely against each other. A good strategy is to bond them through a small number of connection points loosely spreading across the PTFE film (Figure 5.10a). Alternatively, they can be bounded through the edges connecting these points (Figure 5.10b). Both approaches leave the majority of the PTFE film surface free to bounce, maximizing separation distance. In this study, I tested two bonding strategies on the sensor: (1) gluing at the four corners of the PTFE film (Figure 5.11a); and (2) gluing the four edges of the PTFE film (Figure 5.11b). I also tested these bonding methods with different types of glues and found that 3M plastic glue performed the best to hold the entire unit firmly against all the woodworking operations from Study 1.

![Figure 5.10](image)

**Figure 5.10:** Two strategies to bond the copper and PTFE films together: (a) bonding through a small number of connection points and (b) bonding through the edges connecting these points.

Figure 5.11c shows the result of this test. I found that across all the tested frequencies, the sensor was approximately 3db more sensitive when bonded through the corner points than through the edges. Therefore, I chose to use corner bonding in the remaining studies and in the implementation of my final prototype.
Figure 5.11: The prototype used in the bonding strategy test was created by: (a) gluing the copper and PTFE films together through the four corners and (b) gluing the copper and PTFE films together through the four edges. (c) Sensor data of the two bonding

Material Choice for the Electrodes

We conducted a second test to identify the proper material for the electrodes. Note that in my implementation, one of the electrodes also serves as the positive triboelectric layer. Therefore, it is unclear whether electrode material may affect sensitivity. To answer this question, I implemented and tested sensors with electrodes made of four different types of conductive materials: (1) copper film, (2) aluminum foils, (3) carbon coating, and (4) nickel coating (Figure 5.12). All of these conductive materials are low-cost and widely available on the market. Among these options, copper film and aluminum foil were used in previous research in triboelectric nanogenerators [206]. While carbon and nickel coating has not been used for the same purpose, spraying the electrodes directly on the plywood leads to better structural integrity for the sensor.

The result is shown in Figure 5.12e. The strength of the sensor signal decreased in the aluminum foil condition between 150Hz and 300Hz. There was no significant difference between all the other three electrode materials. Considering structural integrity, both carbon and nickel coating would work better for us. I used nickel coating in the remaining studies and in the implementation of my final prototype.
Figure 5.12: The prototypes used in the electrode material test were created using: (a) copper film, (b) aluminum foil, (c) carbon coating, and (d) nickel coating. (e) Sensor data of the tested electrode materials.

Material Choice for the Positive Triboelectric Layer

In the third test, I compare the sensitivity of the current implementation versus an alternative approach with a dedicated positive triboelectric layer, made of polyurethane (PU) and nylon (Figure 5.13b-c). Both of them are commonly used in triboelectric nanogenerators [206]. PU coating has the benefit of better structural integrity for the sensor. My result is shown in Figure 13d. SNR dropped in the PU condition. No obvious improvement could be found with a dedicated positive triboelectric layer using nylon. Therefore, I chose to not change the structure of my current sensor implementation. This allowed the sensor to remain simple to be fabricated.

Figure 5.13: The prototypes used in the positive triboelectric layer test were created using: (a) no additional positive triboelectric layer, (b) polyurethane, and (c) nylon. (d) Sensor data of the tested conditions.
5.2.10 iWood v2 – Makeable Interactive Plywood

Based on the results of the tests, I implemented my second iWood prototype. In this section, I describe my implementation details and a software pipeline for gesture and activity recognition.

Fabrication

We created the electrodes directly on the plywood substrates (610mm × 610mm) using nickel spray paint (Figure 5.14a). The diamond patterns for the two layers were created using acrylic stencils, made using a laser cutter, and were fixed on top of the corresponding substrates using double side tape. Upon the completion of the electrode layers, the row and column connections were created using 2mm wide copper tape (Figure 5.14b). Similar to iWood v1, I attached a PTFE film firmly to an electrode layer and substrate with no space for bouncing. The other layer and substrate were connected using the point bonding strategy with the bonding points separated 50mm apart from each other in the x and y directions (Figure 5.14c). To achieve the best bonding strength, I primed the PTFE film with a plastic glue activator before the layers were put together. I created two copies of the prototype with different thicknesses (12.7mm and 12.7mm) to satisfy the need for different applications (see Section 9). Each prototype costs no more than 25% higher than a regular plywood board of the same size (plywood board: $20, PTFE film: $2, nickel coating: $3).

To facilitate the connection to the electronics and data measurement device, I marked several access points on the plywood to indicate the location of the top (green) and bottom (red) electrodes (Figure 5.14d). The access points can be connected to the electronics in many different ways. For simplicity’s sake, I put a small screw at one of the top and bottom access points and wired them to an Analog Discovery 2. Sensor data was sampled at 1kHz and was streamed to a laptop for processing. Note that with the advancement of technology, all the other electronics components may eventually be integrated into the material [5]. Within the scope of this work, I focused on the sensing component.
Signal Processing

The raw sensor data was first transformed from the time domain to the frequency domain. I then used a low-pass filter at 500Hz to remove the high-frequency components that are unlikely caused by user activities. Additionally, I performed a band-stop filter to remove the harmonic frequencies of powerline noise (i.e. 60Hz and 300Hz). Further, I performed an adaptive background subtraction to remove all the other random noises.

The input gesture or user activity in the signal data was segmented using a coarse-to-fine energy-based sliding window approach. I first employed a 3-second sliding window with a 90% overlap to detect the occurrence of an event of interest. To do so, I calculated the energy of the sliding window by summarizing the square magnitude of its 256 FFT bins. If the energy was higher than a predetermined coarse-power threshold, an event was detected. Otherwise, I moved on to the next window. Upon the detection of an event, I applied a new 0.5-second sliding window with a 75% overlap to look for the start and end of the event at a finer granularity. If the energy of a window rasied above a predetermined fine-power threshold, the start of the event was identified. I then advanced the sliding window until the end of the event was found, based on the drop of energy of the window below the fine-power thread. The thresholds used in my implementation were determined using a pilot study and were left unchanged in my system evaluation.
Featurization and Machine Learning

Using the segmented data, I derived machine learning features in the frequency domain using a sliding window of 512 bins with a 75% overlap. I calculated max, mean, 1st quantile, median, 3rd quantile, h-mean, moment, skew, kurtosis, and standard deviation for each frequency band (256 × 10 values). I also computed the length of segmented data in the time domain. In total, my machine learning model was trained using 2561 features. I used a Random Forest from Scikit-learn with a forest size of 100 and a maximum depth of 30. The value of these parameters was chosen to balance sensing accuracy and model complexity. I ran the classifier on a MacBook Pro.

5.2.11 Demo Project Made of iWood

In this section, I demonstrate the capability of my makeable interactive plywood through some of the common household items made of iWood. I demonstrate how to create a smart table, nightstand, and cutting board using iWood and how new smart home applications can be enabled on these items. All the projects described in this section were created by a volunteer (28-year-old male) with some woodwork experience. The fabrication processes involved all the six operations tested in my makeability study (O1 – O6). All the sensors remained functional in the finished projects, which were used later in my system evaluation to measure the recognition accuracy of my system.

Table

Fabrication: The table consisted of four wooden legs attached to a 1220mm × 610mm × 12.7mm plywood tabletop, which was created using two iWood boards attached side-by-side (O4) (Figure 5.15a). Each leg came with a pre-installed 8mm hanger bolt so pilot holes were created at the four corners of the tabletop using a drill driver (O3) (Figure 5.15b) to allow the legs to be mounted (O2) (Figure 5.15c). To finish the process, the surface of the
tabletop was smoothed using sandpaper (O5) and coated with protective PU to prevent it from water and warping (O6).

**Applications:** The smart table can be used as an extension of the input device on a phone through the detection of simple input gestures. In my implementation, the system can recognize tapping the table using the fingertip, swiping the finger against the table, knocking the table using the knuckle, and slapping it using the palm (Figure 5.18 G1-G4). For example, knocking on the table can activate a smart speaker to listen to a user’s command. Alternatively, slapping on the table can interrupt the response from the speaker if it misunderstands the user’s request (Figure 5.15d).

Further, the smart table can be configured to log the routine activities or work progress of the user and respond accordingly to enable new applications in work, education, or health. For example, in my implementation, the table can sense events like writing, erasing, stapling, dispensing a tape (Figure 5.1b), and sharpening a pencil (Figure 5.15e). By analyzing the duration and frequency of these events, the system can infer higher-level work activities, such as content creation, editing, or organizing documents. For example, content creation can be inferred from a long period of writing. Editing can be inferred from the occurrence of erasing mixed with shorter periods of writing. Organizing paper documents can be inferred from the occurrence of stapling or tape dispensing. This information can be useful to inform the user about their work practices for personal reflection or infer the work psychological state of the user for social facilitation if unusual patterns are observed.

**Nightstand**

**Fabrication:** The structure of the nightstand was more complicated than the table. It consisted of a top panel, a bottom panel, three side panels, four legs, and a drawer. The top panel (500mm × 500mm × 12.7mm) and drawer bottom (310mm × 375mm × 6.35mm) were created using iWood and a jigsaw (O1) (Figure 5.16a). The top panel was attached to the frame and side panels from the corners using four 10mm wood screws (O2) (Figure
Figure 5.15: The fabrication process of my smart table: (a) Gluing two pieces of iWood side by side. (b) Drilling pilot holes at the corners of the tabletop for mounting the legs. (c) Screwing the legs in the pilot holes. Demo applications of the smart table: (d) A user can knock the table to activate a smart speaker and slap it to interrupt its speech. (e) Through the detection of the work activities carried out on the table, such as using a pen sharpener, the system can infer the user’s work practices and detect unusual patterns.

5.16b). The drawer bottom was fixed to the notch of the drawer sides using glue (O4). I also polished the top panel using sandpaper (O5) and painted it in espresso stain to match the color of the rest of the nightstand (O6).

Applications: Similar to the smart table, the nightstand can also be used as an input device to control home appliances remotely. For example, a user can knock anywhere on its top to turn on a light at night (Figure 5.16c). This is helpful, especially for elderly people, as finding a light switch could be hard in the darkness. As an activity sensor, the top panel can detect the user’s routine task of pumping lotion for nighttime skincare. The drawer can also detect and log the user’s bedtime reading habit through the detection of the user taking the book from the drawer and putting it back (Figure 5.16d). If this routine breaks during a busy week before the user’s school project is due, which can be detected through the work activities via the table, the user can be reminded via the phone or this change of behavior can be logged for the user to view at a later time.
Figure 5.16: The fabrication process of my smart nightstand: (a) Cutting the interactive plywood into parts of desired sizes. (b) Assembling the parts using screws and glue. Demo applications of the smart nightstand: (c) A user can knock on the nightstand to turn on a nightlight. (d) Through the detection of the user taking the book from the drawer, the system can detect and log the user’s bedtime reading habit for personal reflection and assistance.

Cutting Board

Fabrication: Like regular plywood, iWood can withstand laser cutting. My volunteer created a custom cutting board (400mm × 600mm × 6.35mm), using graphics editing software and a laser cutter (O1) (Figure 5.17a). No additional configuration was needed for the laser cutter and the machine was set to cut the board using its exact thickness.

Applications: The smart cutting board (Figure 5.17b) can recognize common cooking activities on it, such as chopping, slicing, meat tenderizing, stirring, grating, and rolling a rolling pin. This information can be used to augment cooking experience in many different ways. For example, a progress indicator can be shown on the user’s tablet to provide a better awareness of the time left for their steak to be tenderized or the scrambled eggs to be stirred (Figure 5.17c). The smart cutting board can also find applications in the automatic skill assessment, similar to prior work for working space [20]. For example, the user’s expertise level can be analyzed based on the duration of each activity and overall food preparation time. Such automatic assessment could create opportunities for in-situ feedback, skill-level
evaluation, and skill degradation detection. Most generally, activity recognition through
the smart cutting board could bring richer context-sensitive applications. For example,
interactive recipe apps could work better with the smart cutting board to offer contextual
timing control over the advancement of recipe steps. A video tutorial can be paused or
resumed automatically according to the user’s action detected on the cutting board (Figure
5.17d).

Figure 5.17: (a) My smart cutting board was created by cutting interactive plywood into
the desired shape using a laser-cutter. (b) The finished cutting board. (c) Sensing the
stirring event on the cutting board allows the system to show the time left for the egg to be
stirred through an interactive progress bar. (d) The system automatically pauses a cooking
tutorial video when the user starts tenderizing the meat. Upon the user stops the action, the
system resumes the video.

5.2.12 Evaluation

We conducted an experiment to measure the sensing performance of the table, nightstand
(top panel), and cutting board created in the last section. In particular, I were interested in
measuring how accurate user gestures and activities can be recognized on these items. To
push the limit of my system even further, I evaluated system robustness against individual
and device variances among different users and smart items, which in my case were created
in different form factors with sensors in different shapes and sizes.

Participants

Ten right-handed participants (average age: 27.2, 7 males, 3 females) were recruited to
participate in the study.
Gestures and Activities

My experiment included 12 daily activities ranging from those commonly carried out on a cutting board, such as chopping, slicing, stirring, tenderizing, grating, and rolling a rolling pin, to those that can be carried out on a table or nightstand, such as writing, erasing, stapling, pumping lotion, dispensing a tape, and rotating a pencil sharpener (Figure 5.18). I also included four input gestures, including tapping, knocking, slapping, and swiping in any direction. The activities and gestures tested in my experiment produced vibrations varying in strength and frequency across different users and smart items. While powered devices like a blender can also be recognized through unique vibration patterns, I focused on the activities driven physically by the user to demonstrate the capability of iWood and its potential in many new application scenarios.

Data Collection

Before the experiment started, participants were given several minutes to learn the activities and gestures. During data collection, participants were asked to perform the tasks on the corresponding smart items in whatever way they felt comfortable, either in a standing or sitting position. The order of the activities, gestures, and the 10 repetitions of them were randomized for every participant. Note that not all the activities and gestures are performed on all the smart items. For example, writing may take place on the table or nightstand (e.g., writing a quick note) but not usually on a cutting board. Therefore, the data collected on the table and nightstand only included writing, erasing, stapling, pumping lotion, dispensing a tape, rotating a pencil sharpener, and all the hand gestures. Similarly, the activities carried out on the cutting board are unlikely to occur on a nightstand. Therefore, the data collected on the cutting board (placed on a regular desk) only included chopping, slicing, stirring, tenderizing, grating, and rolling a rolling pin. I also repeated the same set of kitchen activities on the table. Repeating the same group of activities and gestures on different smart items allowed us to investigate how reliable my system can recognize them on different
Figure 5.18: Input gestures and user activities tested in my experiment.

Results

In this section, I report the performance of my system measured using a variety of different ways, including within-user accuracy, cross-user accuracy, mixed-item accuracy, cross-item accuracy, and item identification accuracy.

**Within-User Accuracy:** Within-user accuracy was the measurement of the prediction accuracy on a specific smart item where the training and testing data were from the same
user. For each participant, I conducted a twofold cross-validation, where half of the data was used for training and the remaining data for testing. The overall within-user accuracy was calculated by averaging the results from all the participants for each smart item.

For the four gestures, the system achieved an average within-user accuracy of 93.0% (std: 1.4) on the table and 90.7% (std: 3.9) on the nightstand. In particular, on the table, the accuracy for Tapping, Swiping, Knocking, and Slapping was 91.0%, 94.0%, 94.0%, and 93.0% respectively. On the nightstand, the accuracy for Tapping, Swiping, Knocking, and Slapping was 92.0%, 94.0% 85%, and 92.0% respectively. The major source of errors came from the confusion between Knocking and Slapping. This is because participants performed Knocking sometimes using the knuckles and other times using the phalanges, which had produced signals similar to that of Slapping.

For the activity recognition, the system achieved an overall within-user accuracy of 95.3% across all the smart items. In particular, the system achieved an average accuracy of 94.7% (std: 4.1) on the table, 93.8% (std: 3.8) on the nightstand, and 97.3% (std: 1.9) on the cutting board. Figure 5.19 shows the confusion matrices of the tested activities on the three items. Most activities can be recognized with high accuracy, especially within a smaller subset carried out on the cutting board or nightstand. The recognition accuracy remained high when it came to the table with more activities, suggesting that the within-participant signals were fairly consistent and robust to item variation. The only exception was the rolling pin (83.0%, std: 2.1), which was confused with Stirring more often (92.0%, std: 0.9). Considering the size of my training sample was relatively small in the twofold cross-validation, I suspect that if additional data were collected, accuracy would rebound.

Cross-User Accuracy: Across-user accuracy measured how well a model worked across different users. I conducted a leave-one-subject-out cross-validation by using the data from nine participants for training and the remaining one for testing. My result showed that the cross-user accuracy for gesture recognition was 88.5% (std: 4.0) on the table and 85.0% (std: 6.1) on the nightstand. In particular, on the table, the accuracy for Tapping,
Swiping, Knocking, and Slapping was 85.0%, 86.0% 89.0%, and 94.0% respectively. On the nightstand, the accuracy for Tapping, Swiping, Knocking, and Slapping was 93.0%, 78.0%, 84.0%, and 85.0% respectively. Note that Tapping and Swiping caused more confusion. Based on my observation, their signal could be similar to each other when some participants tended to swipe at a short distance.

For activity recognition, the average cross-user accuracy across all the tested items was 89.9%. In particular, the system achieved an average cross-user accuracy of 87.8% (std: 5.5) on the table, 88.3% (std: 8.8) on the nightstand, and 93.6% (std: 1.9) on the cutting board. Figure 5.20 shows the confusion matrices for these items. With the inclusion of individual variance in the data, the activities with a similar motion began to get more errors. Examples include Erasing vs Grating and Pumping vs Tape Dispensing. For the activities with more distinguishable motions, such as Chopping vs Slicing, the recognition accuracy also decreased. To understand the reason, I examined the data closely and found that some participants were faster than the others when performing the tasks due to individual differences in skill and expertise level. I see it as a strong promise of my system in skill assessment applications. Overall, despite the 5% decrease in comparison to the within-user accuracy, my system still performed reasonably well across different participants. Again, my model was trained with a relatively small sample size. I suspect that with more training data, the accuracy would increase.
Mixed-Item Accuracy: Mixed-item accuracy measured whether a general model can be trained for all three smart items. My general model was trained by including all the data collected from the table, nightstand, and cutting board. The performance of the model was evaluated through (1) within-user accuracy using a twofold cross-validation and (2) cross-user accuracy using a leave-one-subject-out cross-validation.

For the input gestures, the system achieved an average within-user mixed-item accuracy of 88.3% (std: 4.0) and cross-user mixed-item accuracy of 86.6% (std: 5.2). For activity recognition, the within-user mixed-item accuracy was 88.9% (std: 7.2). The cross-user mixed-item accuracy was 84.2% (std: 8.1). Figure 5.21 shows the confusion matrices for the two types of accuracies. While the general model did not perform as well as the item-specific models, it is still encouraging to see that the system remained reasonably accurate. The promise of the general model is that it avoids the need to rely on individual models to be trained separately on each smart item. A single model can be distributed to handle a subset or perhaps the entire inventory of home items. Note that if I removed the most confusing activities, Pumping, Tape Dispensing, Slicing, and Grating, the within- and cross-user mixed-item accuracies rebounded to 91.3% (std: 5.4) and 91.4% (std: 6.0) respectively.

Cross-Item Accuracy: Cross-item accuracy measured if a model worked on a smart item that was different from the one, on which, the model was trained. I measured the
Figure 5.21: The confusion matrices of mixed-item accuracy and cross-item accuracy.

cross-item accuracy by using the data collected from the smart table for training and the data collected from the nightstand and cutting board for testing. For example, the work activities were tested on the nightstand using the model trained on the table. Similarly, the kitchen activities were tested on the cutting board using the model also trained on the table.

My result showed that the cross-item accuracy for gesture and activity recognition was 63% (std: 15) and 64.5% (std:26.2) respectively. The result suggests that my model is not generalizable across the tested items. To understand the reason, I examined the signal data and found that the signals for the same activity varied quite significantly across different items, which, I believe, was primarily due to the difference in the structure of the items.

**Smart Item Detection:** With the data collected on the different smart items, I were interested in investigating if a model could be trained to infer the identity of an item based on the input signal. With a model like this, the system can predict whether an event takes place on a table, nightstand, or cutting board. I measured the item detection accuracy using a twofold cross-validation with the dataset composed of all the gestures and activities. My result yielded an overall accuracy of 90.1. The result suggests that it is possible to identify a smart item based on its unique vibration signature caused by different types of activities and input gestures. This is encouraging even though my experiment only included three items.

While in many usage scenarios, the identity of a smart item will be known to the system, like when a new item is created. In many other scenarios, especially when an existing item...
needs to be reproduced, such self-awareness allows the system to automatically adapt a proper item-specific model to better recognize input or contextual events on that item.

5.2.13 Limitations and Future Work

We present insights I learned from this work, discuss the limitations, and propose future research.

Short-circuiting

My implementation minimizes the chance of short-circuiting to around 5% but when a short circuit occurs, the sensor malfunctions. This can be an issue, especially after the completion of the assembly of a smart item. To address this problem, my immediate next step is to explore solutions to eliminate the short-circuiting issue. Note that even though I advocate for makeability in this project, it is plausible that not all interactive materials could be made 100% makeable. As such, tools need to be developed to allow easy diagnosis, debugging, and visualization of the system status of interactive materials. Unlike the existing circuit debugging tools, which require the users to have certain technical backgrounds to use, the diagnostic tools for interactive materials should be designed for the workers of the material to fit their background and skill set.

Composition of two or more interactive plywood parts

My current work investigated smart objects and items composed of a single piece of iWood or two disconnected ones. In many usage scenarios, multiple interactive plywood boards may need to be connected to create a larger item, such as a floor, thus new challenges will arise. I see that tools will need to be developed to enable the easy creation of iWood clusters to satisfy users’ need to connect the sensors in series or parallel.
Sensing 2D information

When multiple pieces of iWood are arranged in a 2D space, coarse-grained 2D information can be sensed if the sensors are connected in parallel. This way, each iWood serves as a pixel of a larger sensing area. An alternative approach is to enable 2D sensing on individual iWood sensors. This will allow finer-grained 2D information to be sensed, which could enable a broader range of new applications. As a part of my plan to continue this work, I will extend the current implementation of iWood to sensing 2D information. This will involve redesigning the structure and electrode layout of the sensor.

Beyond sensing

An important long-term future work in this line of research is to enclose output (e.g., actuation), power, and other computing components in iWood. While computational material as a self-contained package including all the necessary electronics is a future less likely to happen in the next 10 years, mid-ground approaches do exist to make the reality come earlier. As a part of my current and future research, I am investigating ways to encapsulate computing and power components in wood connectors, such as screws, and use them as an edge computer to control iWood in the wild.

How iWood will influence people’s work practices?

While my current research focuses primarily on the technical aspect of interactive plywood, the outcome opens opportunities to better understand how iWood or interactive materials, in general, could influence people’s work practices in everyday contexts. I see that iWood could be used by the community as a platform to study how the addition of sensing capability in plywood may influence woodworkers’ work practices, their ability to create new things, and their thoughts about their job. In a broader sense, I see that new knowledge can be developed to understand how computational materials may influence the workforce in both tech and traditional industries.
5.2.14 Conclusion

Through a new interactive plywood prototype, I explored how a smart physical world comprised of wooden furniture and kitchen items could be created in the future using the established methods and woodworking operations, by which, the current physical world is created. I advocate the notion of makeability as an inescapable consideration when developing interactive materials. I argue that in the new era of ubiquitous computing, interactivity and computing should be treated as materials’ digital properties, which adds to the already existing natural properties, such as stiffness and conductivity. To smoothly blend the digital and physical worlds, I propose that a material’s digital properties should be inheritable by anything that is made of this material. I demonstrated through iWood that its ability to sense vibration can be inherited by three household items made of it, including a table, nightstand, and cutting board. With vibration sensing, these items can now detect a variety of user activities and input, while still being able to largely preserve the look and feel of their non-computational counterparts. I believe that my research may serve as important groundwork for the future development of computational materials and smart environments.
Chapter 6

Conclusion

6.1 Thesis Contributions

In this dissertation, I have explored the approaches to make interfaces of computational materials more sensitive and makeable. These approaches are demonstrated through five research projects, which transform computational textiles to be sensitive to touch and touchless inputs, as well as contextual information such as objects and activities, and also revolutionize computational plywood to be more makeable against woodworking operations.

Technically, my research builds new computational materials from scratch, which contributes (1) new designs of electronic and material structures (e.g. textile antenna), (2) machine simulations and experiments for structure optimization, (3) novel fabrication methods and (4) sensing and machine learning algorithms for input recognition.

More broadly, my thesis offers intellectual merit in the domains of human-computer interaction, sensing technologies, and ubiquitous computing. It provides a unique and original perspective on how computational materials, which are sensitive to a wide variety of user inputs and contexts and makeable against a wide range of fabrication operations, can better support the creation of smart everyday objects on a large scale. It has the potential to be a fundamental literature in realizing the vision of ubiquitous computing.
6.2 Future Directions

My long-term research vision is to empower people to easily create smart everyday objects on a large scale using makeable computational materials and established methods. In the following section, I outline the four critical components of computational materials necessary to achieve this long-term vision.

6.2.1 Output

Throughout my dissertation, I mainly discussed the input of computational materials because existing output modalities in the environment, such as sounds from speakers and displays from TV, are quite sufficient for simple interactions, as demonstrated in the research projects. However, this does not mean that the output of computational materials is unnecessary. Previous work has shown the importance of hidden displays on computational materials for ambient computing [191]. Additionally, haptic feedback is widely considered to be critical for users to confirm touch interactions with computational materials. Thus, one future direction extending from this thesis is to investigate approaches that can embed output capabilities, such as displays and haptic feedback, into computational materials.

Another research direction in this field is to study how responsive computational materials should be and what role they would play in a smart environment. For instance, if the computational materials sense unhealthy user behavior, should they immediately provide feedback, or should we develop some output strategies to inform users? While there is related work in the literature [207, 208], they are not yet validated in an interactive and responsive system. Thus, it is still worth exploring this direction.

6.2.2 Computing and Networking

In my dissertation, I did not address the integration of computing and networking components into computational materials, as my research focused on resolving interface chal-
To create prototypes, I used expensive, bulky, and power-hungry controllers and circuit boards to manage computing and communication aspects. This approach increased costs, raised form factor issues, and caused significant battery problems. Although advancements in material science may eventually solve these problems [5, 6], it is still worthwhile to explore practical solutions to integrate computing and networking components into computational materials.

To address these issues practically, there are a couple of research directions inspired by my dissertation. First, we could integrate miniature RF microchips, such as NFC/RFID chips, into computational materials. These chips are tiny, inexpensive, and batteryless, and can be easily hidden in the materials. They also have basic capabilities for computing and wireless communication. By utilizing them to operate the interfaces of computational materials that I develop in my thesis, we could solve the problems of using bulky devices for computing and networking capabilities of computational materials.

Another approach to address this issue is to connect computational materials to external devices after the object has been made from them. This approach does not require the initial integration of computing and networking capabilities into computational materials, which reduces the fabrication cost and e-waste associated with them. For example, computing and networking components could be encapsulated in a wood connector, such as a screw, which could function as an external device that is inserted into the computational wood using established fabrication methods to activate it after the wooden object is created. In this way, the computational wood does not inherently possess computing and networking capabilities, but it can be operated by connecting it to other devices.

While both of these research directions appear promising, it is worth noting that the optimal solution to this issue may vary depending on the type of material. This is because different types of materials are used to create different objects, and the specific needs for how computing and networking components are integrated into computational materials also vary accordingly. For example, computational wood is often used to make large ev-
everyday objects such as furniture (e.g. tables) and building structures (e.g. floors). In these cases, the second approach may be more effective, as the objects are stationary and their computational materials can be easily connected to an external workstation through physical wires for computing, networking, and power. However, this may not be as effective for computational textiles, as textiles are used to create wearable or portable objects, such as clothing and cushions. In these cases, the first approach (i.e., RFID) may work better because the computational textile could initially include the capabilities of computing and wireless communication. This would mean that users would not need to worry about how to connect these textiles to an external workstation or circuit boards.

6.2.3 Makeability and Scalability

In Chapters 9 and 10, I discuss the importance of the makeability of computational materials and demonstrate a makeable interface through the iWood project. I expect that this dissertation, with its motivations and research methods, would serve as inspiration and a guide for other researchers to pursue this goal together. After all, there are still numerous sensing interfaces that require redesign to achieve makeability. For example, the sensors developed in Chapters 4, 5, 7, and 8 are not yet makeable because their sensor structures are different from the structure in iWood. Moreover, in a broader sense, makeability encompasses not only the interfaces but also the computing units and all other components in computational materials. How to maintain the makeability when designing those components is also a significant challenge. Despite these challenges, the rewards of creating makeable computational materials are immense, as it is key to democratizing and making computing technology scalable in our environments.

It is worth noting that while the ideal goal is to create computational materials that can be handled, operated, and processed like regular materials, I realize that it may be a daunting challenge to achieve this without compromise. In fact, through our interviews with woodworkers and fashion designers, we have come to understand that tradeoffs, such as
requiring workers to take an additional step to remediate damaged computational materials, are acceptable. However, the key is whether they have the necessary skills to perform the remedial actions and how the remedy affects their design and implementation process. From this point of view, we may find that makeability is not only a technical issue but also a psychological one. For example, how do we evaluate the makeability of computational materials with workers if they are not perfectly makeable? This intersection between technology and psychology is at the core of HCI research.

6.2.4 Software Platform

While the previous sections have highlighted the future directions of hardware aspects in computational materials, it is also important to take into account the software part. Just like the development of personal computers and mobile phones, a novel software platform is needed to empower designers and developers to build diverse applications for everyday objects and environments made of computational materials. This endeavor may require the creation of an AI system that is robust enough in real scenarios, as well as a software tool that is low-code and connected to the physical world. These components are essential for facilitating application development, which I discuss below.

Robust AI System

Regarding the AI system, sensing algorithms and models are fundamental components. Although they were developed in my dissertation and evaluated to be robust enough in a lab-controlled experiment, their performance in the real world remains uncertain. For instance, the objects used by a user may differ significantly among individuals and may be much more diverse than the object set evaluated in Chapters 7 and 8. Therefore, it is expected that the sensing algorithm may not perform the same in the real world as it did in the experiment. Similar issues arise in activity recognition and other sensing technologies in the HCI and Ubicomp domains. Therefore, ensuring the robustness of AI systems in
practical scenarios remains a significant challenge.

To overcome this challenge, there are several promising research directions. One approach is to synthesize real data collected from experiments with a large dataset simulated from physical engines. This approach could improve the robustness of algorithms or models because the large dataset provided by the simulator will include many diverse tasks, enough to mitigate the over-fitting problem. However, it still requires more research efforts to validate how the model can transfer the parameters learned from the simulated data to accommodate real data collected in the experiment. Another research direction is to make the model simple and customizable for users. For instance, the model could initially recognize only a few object categories (e.g., less than 5) and be trained further if the user provides more data for their own objects. Or we could allow developers to customize the model to specifically interpret objects or activities based on the composed objects (e.g., a kitchen table specifically sensing food ingredients and cooking activities). All of these are potential research areas that can address these issues.

**Low-code Software Tools**

In addition to a robust AI system, it is also necessary to build a low-code software tool that enables developers to control logic flows between computational materials or objects made of computational materials. For example, consider a smart home application that alerts users with haptic feedback when a smart table detects an unhealthy combination of ingredients. To develop such an application, we need a tool like MakeCode [209] that allows developers to easily set up logic flows between smart everyday objects made of computational materials. However, the challenge here is how to develop a software tool that supports a wide range of everyday objects made of computational materials. Unlike current software tools like Xcode [210], which ensure application consistency across different devices, we need a fundamentally different approach to achieve the same effect for diverse computational materials and composed objects. This is a potential research direction worth
exploring.
Bibliography


Analog device, 2023.


[100] Estefanía Crespo Bardera, Matilde Sánchez-Fernandez, Luis A. Talegon, and Adrián Vega Delgado. Feasibility of a wearable textile antenna hub based on massive
MIMO systems. In 2016 18th Mediterranean Electrotechnical Conference (MELECON), pages 1–6, April 2016. ISSN: 2158-8481.


191


