COMMUNICATION AND SENSING TECHNIQUES FOR SMART, SEAMLESS HUMAN-ENVIRONMENT INTERACTIONS

by

VIET H. NGUYEN

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Marco Gruteser

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2019

ABSTRACT OF THE DISSERTATION

Communication and Sensing Techniques for Smart, Seamless Human-Environment Interactions

By Viet H. Nguyen

Dissertation Director:

Marco Gruteser

Since its birth over 70 years ago, the computing area has gone over several generations, with significant reduction in size and cost. We are now entering a new generation of computing, called *ubiquitous computing*, where many small computers (smartphones, tablets, sensors, actuators, etc.) are placed on user's body or in the environments and provide many useful services to users. However, current devices and communication methods have not been able to provide fully smart and seamless interaction between users and the environments yet: either users are required to explicitly give attentions to devices and give them instructions to run, or environments need to be equipped with additional infrastructure, often expensive or inconvenient, to monitor user's presence and activities. Therefore, the next generation of computing requires devices that implicitly align with user's intention, obtain necessary information from environments as well as implicitly control them, and smart environments with minimal infrastructure setup to monitor user presence and behaviors.

The goal of this research is to propose novel communication and sensing methods to enable such requirements. In particular, the proposed solutions include: (i) TextureCode, a flicker-free high-speed screen-camera communication technique to help smart glasses equipped with cameras obtain useful information from video stream on electronics displays, (ii) a body-guided communication technique that are used for authenticating users with devices and objects on every single touch interaction, (iii) EyeLight, a sensing system

based on visible light to provide indoor occupancy estimation and room activity recognition services, and (iv) HandSense, an on-hand capacitive coupling-based sensing system for recognizing micro, dynamic finger gestures suitable for controlling head-mounted devices. We believe these systems provide users with more seamless interaction with surrounding environments: as the environment-user interactions implicitly align well with user's intention, data exchange, sensing, or authentication happens in the background without interruption to user's workflow.

Acknowledgements

The PhD process is long, but to me it is also a wonderful journey in which I have the luxury of contemplation of interesting ideas, challenging myself with developing hypotheses, building systems, conducting experiments, and presenting research findings. It helped me understand the joyful life of a scientist: not only can one learn new things everyday, but also can freely challenge those facts, explore new knowledge, and then share it with research communities.

My great experience owes a lot to all the people who played signficant roles in my life all these years. I would like to express the deepest gratitude to my advisor, Professor Marco Gruteser. It is amazing how much I have learnt under his guidance during all these years. Marco has taught me how to identify research topics from practical problems. He balanced my interest in technical details with his clearly thought out vision of research projects. I have also learnt from him to set the bar high for research quality and research delivery to the audience. I am proud when looking back to all the projects I have done during my PhD, and those efforts were surely not without Marco's guidance. Thank you, Marco, for being the best advisor any PhD student could hope to have; you will always have my best wishes.

I am also grateful to Professor Rich Howard for his guidance in many of my projects. He was generous in time when I approached him with physics and electronics questions that are beyond my knowledge. He helped me see the value of working from the first principles, breaking a complex system into much simpler components for understanding what is going on. His infectious enthusiasm in science and his incredible outlook on life are among the things I treasure the most from my PhD years.

I want to thank Professors Yingying Chen, Wade Trappe, and Tam Vu who gladly accepted to be the committee members of my dissertation defense. I also wish to thank all my collaborators: Ashwin Ashok, Narayan Mandayam, Kristin Dana, Wenjun Hu, Yaqin

Tang, Eric Wengrowski, Mohamed Ibrahim, Siddharth Rupavatharam, Minitha Jawahar, Shubham Jain, Hoang Truong, Phuc Nguyen, and Luyang Liu. It has been great experience working with them.

I also want to thank all Winlab members for their help in formulating ideas during casual discussions, conducting experiments and providing feedbacks for my PhD work. I would like to also thank all my wonderful friends at Rutgers University, especially Tuyen Tran, Hai Nguyen, Long Le, Binh Pham, Trung Duong, Tuan Tran, and Hai Pham for making my Ph.D student life enjoyable.

Finally, I would like to thank my parents, Mrs. Hien Hoang and Mr. Hung Nguyen, my sister Trang Nguyen, for their continued support and understanding all these years. I know they miss me very much and wish me to come home, but still they encouraged me to follow my own path. Special thanks to my wife Trang Nguyen, who has always been by my side, being a well-rounded person to support me when I am lost in thoughts, and my son Theodore Nguyen, whose growing curiosity about everything in life is an adorable thing to see everyday and usually a great source of inspiration for my research. This dissertation is dedicated to them.

Table of Contents

Abstract ii

Acknowledgements iv

List of Tables ix

List of Figures x

1. Introduction 1

1.1. Existing wearable devices and ambient sensing techniques 2

1.2. Research challenges 4

1.3. Design principles 5

1.4. Thesis contributions 6

2. High-Rate Flicker-Free Screen-Camera Communication with Spatially Adaptive Embedding 9

2.1. Introduction 9

2.2. Flicker perception for embedded screen–camera communication 12

2.2.1. Frame rate 12

2.2.2. Image content 13

2.2.3. Saccades 14

2.2.4. Viewer's field of view 14

2.2.5. Hints for code design 14

2.3. Spatial-Temporal Embedding 15

2.3.1. Temporal embedding 15

2.3.2. Spatial embedding based on texture analysis 16

2.3.3. Superpixels 18

2.3.4. Receiver and decoder . . . 19
Frame perspective correction and spatial block division . . . 20
Decoding algorithm . . . 20
2.4. Implementation . . . 21
2.5. Evaluation . . . 22
2.5.1. Communication Performance of TextureCode . . . 24
2.5.2. Comparison of TextureCode with prior work . . . 25
Perceived flicker . . . 25
Goodput and BER . . . 27
2.5.3. Microbenchmarking . . . 28
Communication Range . . . 28
Maximum Transmit Rate . . . 28
2.6. Related Work . . . 29
2.7. Conclusion and Future Work . . . 30

3. Body-Guided Communications: A Low-power, Highly-Confined Primitive to Track and Secure Every Touch . . . 31
3.1. Introduction . . . 31
3.2. Threat Model and Background . . . 34
3.2.1. Threat Model . . . 34
3.2.2. Existing Wireless Technologies . . . 34
3.2.3. On-Touch and On-Body Communication . . . 37
3.3. Body Guided Communications . . . 38
3.3.1. Challenges with employing body communication methods . . . 39
3.3.2. Double capacitively coupled communications . . . 40
3.4. Touch authentication token design . . . 43
3.4.1. Wearable Design . . . 43
3.4.2. Receiver Design . . . 46
3.4.3. Transceiver Design . . . 47

3.5. System implementation . . . 48
3.5.1. Low power token . . . 48
3.5.2. Token for COTS touchscreens . . . 51
3.6. Performance evaluation . . . 52
3.6.1. Difficulty of Eavesdropping . . . 52
3.6.2. Per-touch authentication/identification . . . 57
3.6.3. Power consumption . . . 59
3.7. Discussion and future work . . . 60
3.8. Related work . . . 61
3.9. Conclusion . . . 63

4. Light-and-shadow-based Occupancy Estimation and Room Activity Recognition . . . 64
4.1. Introduction . . . 64
4.2. Background and Related Work . . . 66
4.3. EyeLight design . . . 69
4.4. Tracking Algorithms . . . 73
4.5. Room Activity and Occupancy Recognition . . . 77
4.6. EyeLight prototype and testbed . . . 79
4.7. EyeLight evaluation . . . 79
4.7.1. Light barrier crossing detection accuracy . . . 80
4.7.2. Localization error . . . 82
4.7.3. Room Activity Recognition and Occupancy Estimation . . . 82
4.7.4. Microbenchmark experiments . . . 84
4.8. Discussion and Conclusion . . . 86

5. Capacitive Coupling-based Micro, Dynamic Finger Gesture Recognition 88
5.1. Introduction . . . 88
5.2. Background . . . 91
5.2.1. Existing finger gesture recognition techniques . . . 92

5.2.2. Capacitive sensing . . . 93
5.3. HandSense overview . . . 94
5.4. Design of on-glove electrodes . . . 97
5.5. Design of the capacitive profiling system . . . 99
5.5.1. Transmitter and receiver design . . . 99
5.5.2. Estimation of received signal amplitude . . . 101
5.6. Micro dynamic finger gesture recognition . . . 103
5.6.1. Recognition of different types of finger interactions . . . 104
5.6.2. Neural network-based gesture classification . . . 106
5.7. Evaluation . . . 107
5.7.1. CapProfiler prototype . . . 108
5.7.2. Gesture set . . . 108
5.7.3. Data collection and preprocessing . . . 109
5.7.4. Gesture recognition performance . . . 110
5.7.5. Microbenchmarks . . . 112
5.8. Limitation and Discussion . . . 113
5.9. Related Work . . . 114
5.10. Conclusion . . . 117

6. Conclusion . . . 120
6.1. Summary of contributions . . . 120
6.2. Looking ahead . . . 121
References . . . 122

List of Tables

1.1. Communication and Sensing methods in this thesis. 6

2.1. The screenshots of some test video sequences. 23

2.2. Summary of goodput for the four systems: InFrame++, HiLight, TextureCode and Hybrid system. **S**: Static scenes, **D**: Dynamic scenes 26

2.3. Comparison of Maximum Transmit Rate, which is normalized for the video display frame rate and the number of blocks per video frame. In InFrame++, *bitsPerBlock* is the number of bits per encoded block. In TextureCode, *encodedPercentage* is the percentage of encoded regions over the whole video frame. 29

3.1. Comparison of existing communication methods. 35

3.2. BER vs. distances (received power at each distance is also recorded). 54

4.1. Room activity and occupancy categories 78

5.1. Full gesture set used in our experiments. Note that the illustrations do not include the hand glove. 110

5.2. Classification performance of different neural network-based methods. . . . 110

5.3. Current drawn in each component in our CapProfiler prototype. 113

List of Figures

2.1. State of the art methods for screen–camera communication compromise either visual obtrusion or goodput. TextureCode selectively embeds in textured regions of images to reduce flicker, but still achieves superior goodput. . . . 10
2.2. Signal amplitude experiment. 13
2.3. Performance of flicker perception for different video samples 13
2.4. The block diagram of the different components in TextureCode. 15
2.5. Illustration of the encoding method. 16
2.6. Texton Method of Representing Textured Regions 18
2.7. Texture encoding method (the pixels inside the blue regions are encoded). The texton analysis avoids encoding in plain texture area in the video frame, such as the road, the sky, etc. while encoding in the high texture area, such as the cars, the buildings, etc. The superpixels method then further separates encoded blocks (avoid boundary effect), and also aligns their boundaries to the existing edges in the video frame. 20
2.8. Experiment setting 24
2.9. Transmit rate and goodput performance. (a) Dynamic scene, (b) Static scene. 25
2.10. Comparison between systems. 26
2.11. BER and goodput vs. distance. 28
3.1. Magnetic coupling. 36
3.2. Different coupling types in IBC. 39
3.3. Body-guided communication method: Channel modeling. 41
3.4. SNR at the intended receiver vs. at an adversary on air for different wearable electrode configurations. 43
3.5. Wearable design. 44

3.6. SNR received at the receiver for different form factor positions and different touch scenarios. 45
3.7. SNR difference between touch and no touch for different touch interaction scenarios. 46
3.8. SNR received at the receiver for different frequencies. 47
3.9. Transmitter prototype. 49
3.10. Touch receiver. 50
3.11. Signal received from the receiver board. 50
3.12. Received signal at different distances from the wearable token (wristband form factor). 52
3.13. Touch-based eavesdropping. 54
3.14. Intended receiver's SNR advantage over the adversary. 54
3.15. Received signal vs. distance on arm. 55
3.16. Touch recognition rate vs. transmission power. 56
3.17. Decoding success rate vs. touch duration and code length. 56
3.18. Decode rate vs. transmission rate (COTS receiver). 56
3.19. BER vs. different users. 58
4.1. Overview diagram of components in EyeLight. 70
4.2. Receiver. 71
4.3. Raw readings from one receiver. 74
4.4. Virtual light barrier crossing detection. 76
4.5. EyeLight testbed. There are 6 light nodes with distance between adjacent pair is 2.5m. The room has a central table, a number of chairs, and a projector screen. 80
4.6. The distribution of different activities and occupancy categories in the dataset. 81
4.7. True positive rate and false positive rate of delta detection algorithm for all different sensors. 81
4.8. CDF of localization error for three cases: 1/ using only spikes detection, 2/ using only delta detection, and 3/ combined detection. 82

4.9. Confusion matrix for activity identification in conference room. Total size of the dataset is 102889 feature vectors, corresponding to 28.5 hours. 83
4.10. Confusion matrix for occupancy estimation in conference room. Total size of the dataset is 1710 feature vectors, corresponding to 28.5 hours. 83
4.11. (a) Delta values for different distance between two nodes. (b) Location median error for different number of nodes. (c) Delta values for different ambient light settings. (d) Delta values for different types of floor carpets.(e) Types of carpet in (d). (f) Effect of lamp shade. 85
5.1. **HandSense concept**. While Augmented Reality head-mounted devices start to find applications in areas like manufacturing, repair and maintenance, providing inputs for these devices with low-effort from users remains a challenge. HandSense offers an always-available, user-friendly dynamic, micro finger gesture recognition system for these devices. 89
5.2. System overview 95
5.3. Electrode placement. 97
5.4. Received signal at 100kHz in one-electrode vs. two-electrode designs. Here d is the distance between the two fingers during its transmitting-receiving session. 98
5.5. Analog receiver frontend 100
5.6. Measurement methods for estimation of received signal amplitude. 101
5.7. Illustrations of finger interactions recognized by HandSense. 104
5.8. Received signal vs. distance (Thumb to index finger) 105
5.9. Prototype. 108
5.10. Confusion matrix of finger gesture classification using three neural network-based models. 118
5.11. Effect of the measurement rate on classification performance. 119
5.12. Different glove. 119

Chapter 1

Introduction

In his landmark paper "The Computer for the 21st Century", published in 1999, Mark Weiser, a researcher at Xerox PARC, described his vision of Ubiquitous Computing: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [1]. This computing paradigm aims to remove computing devices as the barriers between humans and their environments, thus enables human-environment interaction to be more intuitive. In recent years, with the advances in many fields such as Wearable Devices, Smart Home/Building, Smart Citites, we are moving closer towards this vision of ubiquitous computing. For example, wearable devices, with their ability to sense different biosignals, offer users the convenience of day-long wellness tracking. Smart homes allow users to control the indoor heating, lighting system from any place. Augmented Reality head-mounted devices start to find applications in several industries (manufacturing, repair and maintenance, health care, etc.) with their ability to overlay digital information on the real world.

However, a closer look at the landscape of computing devices, as well as communication/sensing methods, reveals that there are still several missing components in realizing the "invisible" aspect of Ubiquitous Computing. *First*, users still need to give attention to devices or explicitly give them instructions to run. To get detailed information about a promotion advertisement displayed on an electronic display, a user needs to get his phone out of his pocket to capture a QR code that stores the additional information. A smart indoor AC system, such as a Nest thermostat, still needs to be controlled by a smartphone interface, instead of a system capable of automatically detecting human presence and activities. *Second*, interactions with current smart devices still introduce interruption to user's

workflow. Personal devices like smartphones, smartwatches usually store sensitive user information, so they need user authentication; however, the common authentication methods such as PIN code or passcode are cumbersome since users need to input them after each idle period. To control an Augmented Reality head-mounted device, a user still needs to either tap on device touch pad or perform in-air hand gestures in front of the device camera, thus preventing the absolute hand-free, interruption-free workflow.

Third, some ambient sensing techniques still require complex infrastructure setup. Common sensing techniques, including cameras, wireless sensing, infrared motion sensing, require installation of additional sensors in the environments, and deployment of a large number of these sensors to cover an indoor space would incur significant extra cost. *Fourth*, there are still security and privacy issues in some of the current human-environment interactions. Cameras being used for monitoring indoor user activities can cause discomfort as users increasingly concern about being under surveillance, especially in residential spaces. For authentication of users with devices, some methods based on wireless signal (Bluetooth, NFC) are susceptible to adversarial interception, such as eavesdropping and man-in-the-middle attacks.

This thesis proposes communication and sensing methods, together with systems and devices, with particular focus on the above problems for realizing a truly ubiquitous and invisible computing for users. In this chapter, we will first look at existing wearable devices and ambient sensing techniques for ubiquitous computing in Section 1.1, and several challenges in ubiquitous computing systems in Section 1.2. Then we will lay out some design principles that guide us in developing our communication and sensing techniques in Section 1.3. Section 1.4 describes in details the contributions of this thesis.

1.1 Existing wearable devices and ambient sensing techniques

Current computing devices usually have human attention problems. Let's take smartphones as an example. Although nowadays smartphones are convenient devices with more and more powerful features packed into their small form factors, they require explicit attention from us and bar us from real interaction with people and environment surroundingus. In some

cases, the lack of attention to the surroundings can lead to fatality, such as distracted driving. To realize Mark Weiser's vision of invisible computing, the next generation of devices and communication methods should have seamless integration into the normal interaction between users and environment. There are two major categories of research works on this front: *wearable computing* and *ambient sensing*.

Wearable Computing. The first approach to Ubiquitous Computing is using devices worn on users, since the close proximity to the user body would enable many human-centric contextual sensing opportunities. Currently the most common use of wearable devices is in the form of smartwatches (e.g. Apple Watch) and smart wristbands (e.g. Fitbit, Jawbone). These devices use built-in sensors to provide users day-long activity and wellness tracking. Research has shown other uses of these wrist-worn devices, such as recognizing hand gestures [2,3], recognizing objects being touched [4], tracking user's hand in space [5], detecting eating and smoking behaviors [6], tracking driving behaviors [7]. There are other wearable devices in research, including smart textile, smart rings, smart shoes, head-mounted devices for Augmented Reality, etc.

This thesis looks at these wearable devices from a different perspective: they are convenient gadgets that almost always accompany users and can help them interact with the environment in more effective and seemless ways. The thesis offers different communication mechanisms between on-body devices and in-environment ones that are fast and align well with user's intentions: screen-to-camera communication for information retrieval and body-guided communication for user authentication.

Another aspect related to wearable devices in this thesis is to design a wearable, low-complexity sensing system for finger gesture recognition, which users can use to control head-mounted devices in non-hand-free working environments. Augmented Reality head mounted devices are now bringing benefits to different working spaces as they provide better visualization with their ability to overlay digital information on physical world. An always-available, low effort interface, which is based on small finger gestures, would be beneficial for these devices as it helps reduce interruptions to the user's workflow.

Ambient Sensing. The second approach to Ubiquitous Computing is not to use any devices worn on user at all, but instead embedding computation and sensing devices into the

environment to sense humans by their different characteristics and then react in context-sensitive ways. There are three broad categories in this class: *camera*, *RF*, and *light*. The approach using cameras [8–10] requires capturing raw images and video, therefore raise concerns about privacy risk involving leaking of sensitive images. RF-based activity sensing [11–13] utilizes available indoor RF devices, such as WiFi routers, to sense user presence and activities. Visibile Light Sensing is another approach for the indoor sensing problem. For instance, StarLight [14] equips an office room with LED lights on the ceiling and photosensors on the floor to reconstruct human skeleton of a user in the room.

This thesis proposes an easy way to embed sensing devices into the environment without complicated infrastructure deployment. We present a smart environment based on existing indoor light infrastructure, since light is ubiquitous in human civilization. It also has advantages over camera-based monitoring system, since people are less comfortable when being monitored by cameras.

1.2 Research challenges

There are several challenges when realizing the vision of "invisible computing", as described below.

Unobtrusiveness. Users should not be bothered with the devices communicate with each other, but the devices should intelligently couple to the user's intention. The challenge is the interactions between users and their environments usually have very short duration (e.g. a quick glance, a short touch on a device), and during this short time interval the communication and sensing should not cause any overhead to users.

Understanding human body characteristics. To provide a seamless, intuitive interacting experience between users and their environments, we should understand some aspects of the human perception to the communication and sensing modalities being used. In some cases, there is only a small gap between human perception and machine perception, which requires a careful design to exploit this opportunity. In other cases, some inherent characteristics of the human body, such as conductive tissues, can be utilized for smart sensing and authentication purposes.

Sensing with low resolution data. A common solution for sensing indoor user activities is using cameras. However, this is usually not preferred in residential areas because of privacy concerns. Users can be uncomfortable when knowing that they are being monitored, their workflows or behaviors can be deviated from normal. In this thesis, the two sensing modalities explored in this thesis, in particular visible light and capacitive sensing, have lower resolution sensing data, thus remove privacy concerns for users. However, this low dimension property places a challenge when inferring rich user activities and user gestures. We overcome this challenge by the cooperative mechanism between large number of low complexity sensing elements, as seen in Chapter 4 and Chapter 5.

1.3 Design principles

We follow these design principles in developing the communication and sensing techniques in this thesis, in order to realize the unobtrusive interactions between users and their environments:

Avoid interruption to normal workflow/user behaviors. To ensure smooth experience for users when interacting with devices in smart spaces, it is crucial that the communication and sensing happen in the background without user's notice. We identified several aspects of the current device interfaces that are not interruption-free, including user authentication on devices, interfaces for head-mounted devices, and offered alternative communication and sensing techniques that reduce user workflow interruption.

Minimal instrumentation. Our designs aim to have no or minimal instrumentation in the environments and on human body. In these systems, either users are not required to wear any devices or the wearable devices in use are widely adopted. In particular, for ambient sensing, we propose using the ubiquitous lighting infrastructure in buildings for indoor localization, room activity recognition and occupancy estimation. For wearable devices, we built our system based on commonly worn devices or equipments, including smartwatches, writbands in everyday life, or working gloves in certain industries.

Built-in security and privacy primitives. Security and privacy are two issues that need to be guaranteed in communication and sensing systems involving humans. The two

	Communication	**Sensing**
Visible Light	TextureCode	EyeLight
Capacitive coupling	Body-guided Communication	HandSense

Table 1.1: Communication and Sensing methods in this thesis.

communication and sensing modalities being explored in this thesis, visible light and capacitive coupling, can provide primitives for security and privacy. Visible light communication works on the basis of line-of-sight between devices, thus it is easy to constrain the communication range to be in close proximity (Chapter 2). Taking advantage of the fact that capacitive coupling only works in very short range, chapter 3 presents a capacitve coupling-based authentication technique that prevents eavesdropping attack by constraining the communication channel to be a small region around the user's hand. Sensing with these two modalities is also less privacy obstrusive. The sensing elements (photodiodes in visible light sensing, electrodes in capacitive sensing) provide only low-resolution data that can hardly cause concerns about being surveilled as with using cameras.

High speed. The communication and sensing methods should be high speed, so that within only a brief time of human attention (quick glance, a short touch, a micro gesture, a quick presence), the device should still be able to capture enough information to do useful tasks, such as information gathering, authentication, localization, or activity recognition.

1.4 Thesis contributions

This thesis hypothesizes that very high frequency signal (visible light) and low frequency signal (through capacitive coupling) can be used to augment the human-environment interactions with intelligent services while being unobtrusive to users and having built-in security and privacy aspects. In particular, we embed message in these signals to enable tracking where users look at and touch. We also utilize sensor arrays of these signals for tracking people's indoor movements, as well as subtle finger movements that can be used for controlling Augmented Reality head-mounted devices. In summary, this thesis presents four specific contributions (Table 1.1):

TextureCode: High-speed Flicker-free Screen-Camera Communication with

Spatially Adaptive Embedding [15]. This work utilizes the pervasive use of screens (such as public electronics displays) and cameras, which can accompany users in smart glasses, and offer a high speed communication link from the screens to cameras. This communication link is useful for users to use their wearable glasses to conveniently obtain information from the displays they are interested in. A challenge in designing such a communication link is to improve the communication throughput while maintaining unnoticeable flicker of screens. To achieve both high capacity and minimal flicker, we propose a technique called *spatial content-adaptive encoding*, and combine multiple design dimensions. We develop content-adaptive encoding techniques that exploit visual features of videos on screens, such as edges and texture, to ubobtrusively communicate information. We are able to achieve an average goodput of about 22kbps, significantly outperforming existing work while remaining flicker-free. We present TextureCode in more details in Chapter 2.

Body-guided Communications: A Low-power, Highly-Confined Primitive to Track and Secure Every Touch [16]. This work focuses on developing a secure yet convenient method for user identification, authorization and authentication when users interact with surrounding devices and objects. The motivation is that as the interaction times with each device or object is becoming shorter, the overhead of conventional user identification, authentication and authorization methods places heavy burden on users. Therefore, it is more desirable to do authentication on every single user touch. Our technique to achieve this is based on a hardware token worn on user's body, such as a wristband, which interacts with a receiver embedded inside the object through a body-guided channel established when the user touches the object. We will demonstrate several desirable properties of our solution, including low power, superior resilience to attacks, and robust authentication capability. We discuss Body-guided Communications in Chapter 3.

EyeLight: Light-and-shadow-based Occupancy Estimation and Room Activity Recognition [17]. This work introduces EyeLight, a system embedded in indoor lighting environment to sense the human occupancy and room activities. The system is based on Visible Light Sensing: while previous works require either light sensors to be deployed on the floor or a person to carry a device, our approach integrates photosensors with light bulbs and uses the light reflected off the floor to achieve an entirely device-free

system. We build a prototype system using modified off-the-shelf LED flood light bulbs and install it in a typical office conference room. Evaluation results show that we are able to achieve 0.89m median localization error as well as 93.7% and 93.78% occupancy and activity classification accuracy, respectively. EyeLight will be presented in Chapter 4.

HandSense: Capacitive coupling-based Micro, Dynamic Finger Gesture Recognition. This work aims at the applications of Augmented Reality head-mounted devices in several industries, such as manufacturing, repair and maintenance, healthcare. While these devices provide a convenient digital data overlay on physical world, it is desirable to have a lower-effort, intuitive interface for users to interact with these devices, especially in works where user hands are occupied. This work proposes HandSense, a light-weight, always-available system to recognize a series of dynamic, micro finger gestures that are highly suitable for controlling these head-mounted devices. Chapter 5 gives more details about HandSense.

Chapter 2

High-Rate Flicker-Free Screen-Camera Communication with Spatially Adaptive Embedding

2.1 Introduction

With the pervasive use of screens and cameras, screen–camera communication through QR-code-like tags has emerged in diverse applications from pairing devices to obtaining context from advertisements and other screen content. When placing such codes on screens, they occupy valuable screen real estate, which results in undesirable compromises. Either the visual code replaces most of the imagery on the screen, which usually distracts from the aesthetics of the image or video, or the code only uses a small area of the screen, which leads to less throughput and requires the camera receiver to be closer to the screen. This conundrum motivates embedding such codes into the screen imagery so that the code is detectable with camera receivers but imperceptible for the human visual system.

While there have been a few existing efforts on embedded screen-camera communications, they tend to achieve *either* high throughput but noticeable flicker *or* virtually flicker-free embedding but low throughput, as illustrated in Fig. 2.1. In particular, InFrame++ [18] utilizes the flicker fusion property of the human visual system to embed data. It relies on high screen refresh and camera frame rates to modulate the image at rates faster than the human eye can perceive. It can therefore transmit data at 18 kbps, but noticeable flicker remains. HiLight [19] modulates bits through slight pixel translucency changes, which reduces flicker to unnoticable levels but only supports a low bit rate. The setting is also related to the classic watermarking literature, but only some of the work considers camera capture, usually involving ultra-low data rates of a few bits per second [20], [21]. These are sufficient for digital rights management applications to address movie piracy, but do not meet the capacity and flicker requirements of pervasive screen tags. Overall, existing work

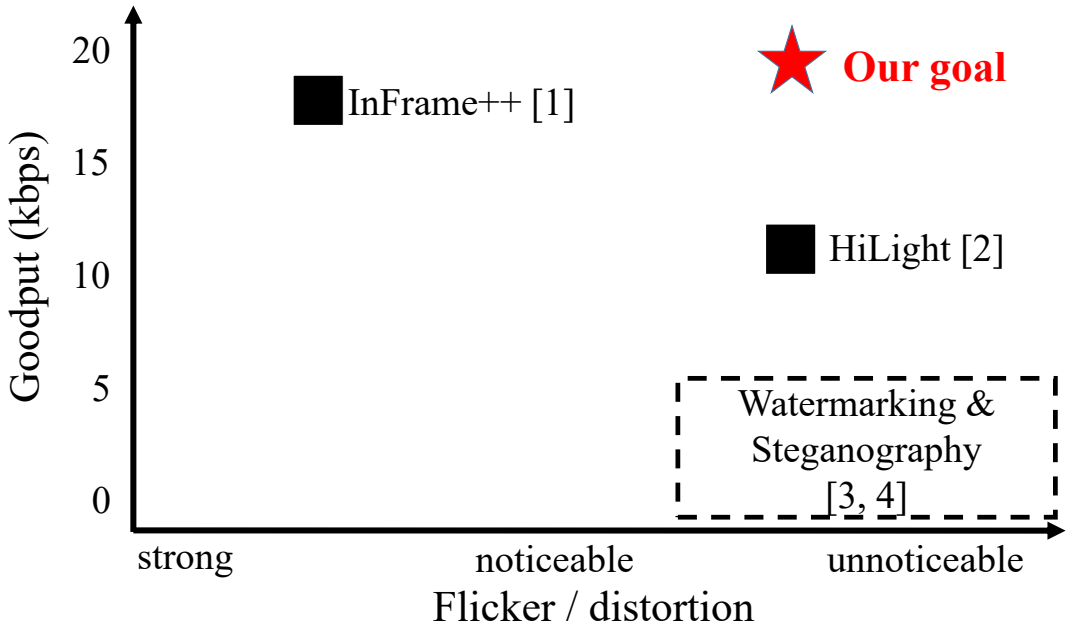


Figure 2.1: State of the art methods for screen–camera communication compromise either visual obtrusion or goodput. TextureCode selectively embeds in textured regions of images to reduce flicker, but still achieves superior goodput.

tends to each explore one technique for embedding, and it is unclear what the limitations are.

In this chapter, we systematically explore psychovisual factors leading to flicker perception and uncover additional dimensions of the flicker-free embedding design space. In particular, we study adaptive spatial encoding in the screen–camera communication channel, which has hitherto remained unexplored.

Conceptually, spatially adaptive encoding in screen–camera systems resembles adapting modulation and coding rates on different streams in a spatially multiplexed precoded Multiple Input Multiple Output (MIMO) radio system. In practice, however, the screen–camera communication channel imposes very different challenges. Radio frequency MIMO often requires precoding because typical MIMO spatial streams interfere with one another and need to be decorrelated for the best encoding opportunities and decoding performance. In the case of screen–camera communication, or visual MIMO, however, the individual pixel-to-pixel links are very directional, and there is little interference between such "spatial" links (neglecting image blur). Instead, the primary challenge is that the modulation and coding

techniques should not only maximize communication performance but also minimize image distortions and flicker for the human observer. In addition, the communication techniques must be robust to noise from the carrier image or video and work without feedback from the receiver, since the screen–camera channel is a one-way channel.

Our work addresses the challenges by exploring several factors for flicker perception and combining corresponding coding opportunities. First, since both flicker perception and receiver noise depend on the visual content of the frame that the information is embedded in, we design a texture-based estimator that determines the suitability for embedding in each pixel block of the screen. This information then governs the choice of modulation and lends to the spatially adaptive approach. It also addresses the unknown channel state at the transmitter, since the texture analysis effectively provides an estimate of receiver noise on each block. Second, the technique aligns the boundary of each encoded region along the existing edges in the video sequences to minimize the visible artifacts caused by encoded messages. Third, akin to earlier work, we also modulate at a rate beyond the critical flicker fusion threshold for most observers but remains decodable with the high-frame rate (slow motion) cameras available in today's smartphones. Finally, we identify a lightweight approach following the same principles and delivering similar performance at a much lower computational complexity.

In summary, the salient contributions of this work are:

- We analyze factors contributing to distortions and the flicker perception of embedded screen–camera communication.
- We identify techniques to achieve spatially and content adaptive embedding. We also show that it is possible to achieve similar performance using a lightweight approach.
- We explore and combine multiple encoding methods to embed information into arbitrary video content without noticeable distortions or flicker.
- We show experimentally that our proposed methods have the potential to more than double the goodput of existing flicker-free screen–camera communication techniques.

2.2 Flicker perception for embedded screen–camera communication

Flicker is a perceptual attribute normally defined for displays, seen as an apparent fluctuation in the brightness of the display surface [22]. Prior psycho-visual studies have revealed various effects in the displayed video that may contribute to the perceivable flicker, such as the frame rate, image content, saccades, and the viewer's field of view.

By inducing brightness changes in a regular video to modulate bits, embedded screen-camera communication can naturally generate flicker. Therefore, we explore how to balance the conflicting goals of embedding bits and avoiding flicker. Where applicable, we perform simple experiments to provide qualitative hints. These follow the same settings as in Section 3.6, and the flicker level is assessed visually by the first two authors.

2.2.1 Frame rate

It has long been known that flicker perception is prominent for luminance fluctuations below 100 Hz [23, 24]. Although this frequency threshold was determined using a single light source, it is still applicable if we consider the modern display as a collection of LED light sources.

In our case, the fluctuation is caused by switching between bits at the same position of the video across frames. Since such bit streams are random, we are constrained by the largest differences between the codewords, the available display refresh rate (up to 144 Hz), and the camera capture rate (up to 240 fps). Given the latter two constraints, we can expect to display at 120 fps. The maximum codeword distance can then be determined accordingly.

We place two uniform grayscale blocks side by side (Fig. 2.2). In each run, the left block has a fixed intensity value x, while the right block's color flips between $x + \alpha$ and $x - \beta$ at 120 fps. Across runs, x varies from 0 to 250 at steps of 25. Experiments show that the color deviation without inducing flicker perception is $\alpha = 2$ and $\beta = 3$. In other words, only very slight color differences between adjacent blocks can be tolerated. This suggests very limited scope for encoding bits directly using pixel intensity changes.

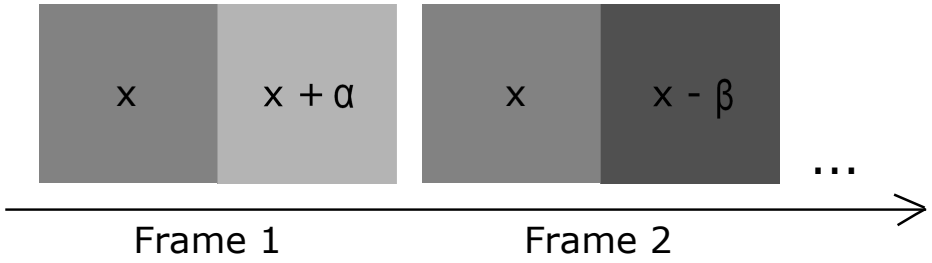


Figure 2.2: Signal amplitude experiment.

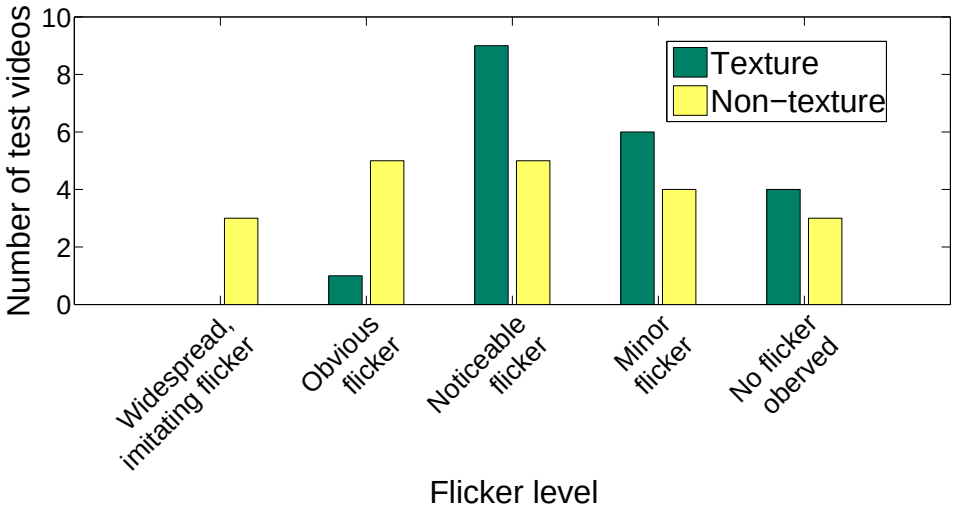


Figure 2.3: Performance of flicker perception for different video samples

2.2.2 Image content

Images of natural scenes often contain many textured regions that we can use in our coding method. It is well known that human vision is sensitive to even small intensity edges [25, 26] and that texture affects the perception of intensity transitions [27]. As a practical consequence of these perception traits, intensity modifications in smooth regions are more likely to cause flicker than textured regions. To take advantage of this flicker reduction, our method adapts to image content by detecting textured regions and embedding message bits within this space.

To qualitatively evaluate the intuition of texture-based embedding, we experiment with 20 videos of varied contents. We divide each video frame into smooth and textured regions (detailed in Section 2.3), and embed bits into the smooth regions only, the textured regions only, or all regions to compare the flicker perception. Fig. 2.3 shows that embedding into

textured regions exhibits the least amount of flicker.

2.2.3 Saccades

Saccades are rapid, ballistic movements of the eyes that abruptly change the point of fixation. In [28], the authors introduced an edge between a white half frame and a black half frame. The colors of the two halves were inverted in rapid succession, and the human subjects still observed flicker artifacts regardless of the switching frequency, even at 500 Hz.

Since it is common to use a block of pixels to encode a bit, we also encounter edges between adjacent blocks of different bits. When the two neighboring blocks are modulated with "different phases", i.e., one block changes from $x+\alpha$ to $x-\beta$ while the other changes in reverse, flicker is noticeable. However, separating the blocks with some distance can reduce or minimize the effect.

2.2.4 Viewer's field of view

In the course of experiments, we also observe that the level of flicker perception depends on the size of the encoded regions in the video and the distance of the viewer from the video displayed. We capture both effects with a single metric, the size of the "viewer's field of view". To measure this size, we use a square block of different sizes for encoding without changing other parameters and view the video from different distances. Results show the smaller area fell into viewer's retina, i.e., the smaller block size or further distance, the less flicker the viewer perceives. This suggests using only small code blocks for encoding and avoiding parts of the image scene that might attract attention.

2.2.5 Hints for code design

We make several observations from the exploration so far. First, the first three factors above suggest opportunities for modulating bits, while the field of view cannot be leveraged easily, since the encoder side has no control. Second, each factor alone offers limited flexibility in modulation. In other words, to control flicker perception in the encoding process, we have to work within a small range of brightness fluctuation, which significantly constrains the code capacity. This is precisely why HiLight and Inframe++ *either* achieves a high goodput *or*

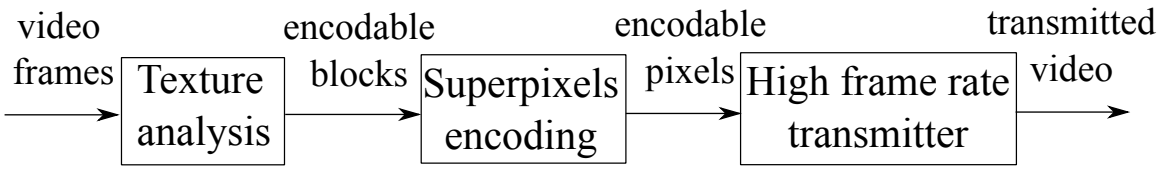


Figure 2.4: The block diagram of the different components in TextureCode.

negligible flicker perception, but not both simultaneously. Third, the first three factors are orthogonal, paving way for combining the corresponding techniques leveraging the factors. Frame rate is a temporal property of the video, whereas the image content and saccades mostly affect the spatial domain. Based on these insights, we design TextureCode to achieve high capacity at negligible flicker.

2.3 Spatial-Temporal Embedding

We exploit these observations of flicker perception and explore schemes that operate both in the spatial and temporal dimensions. We first discuss the temporal dimension through the design of high-frame rate embedding that seeks to operate beyond the human flicker fusion frequency. We then discuss schemes that employ spatial adaptation based on texture analysis to address the image content factor. Finally, we align the boundary of each encoded region along the existing edges in the images, to minimize the effect of visible artifacts caused by encoded messages and to address saccades. This is accomplished through a superpixel encoding technique. We refer to combining these ideas in an approach that we call *TextureCode*. A block diagram of this approach is shown in Figure 2.4. In addition, as we will show in Section 3.6, the video frame content not selected in TextureCode could still be used in other mechanisms to produce a hybrid version with better goodput and no flicker.

2.3.1 Temporal embedding

We apply basic temporal embedding as follows. In our system, we utilize a screen capable of playing video at high speed (120 Hz) as our transmitter. The 120 fps video is created from an original video at 30 fps by duplicating each frame in the original videos to 4 new frames. These 4 new frames are then used to embed messages.

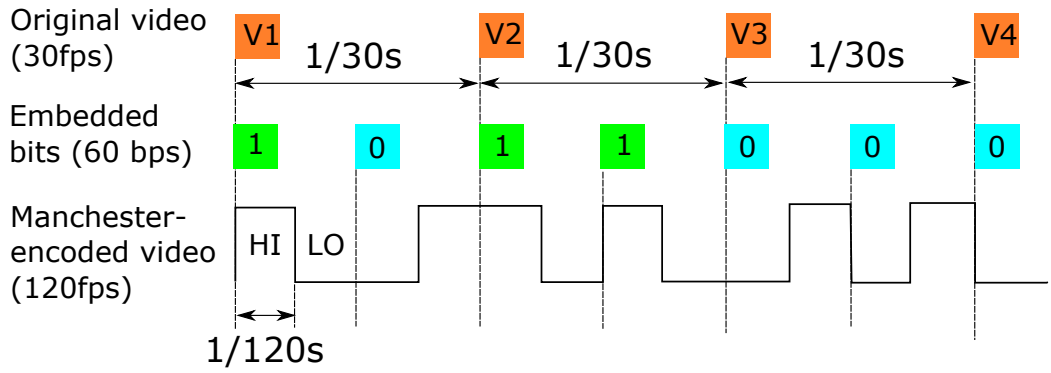


Figure 2.5: Illustration of the encoding method.

Figure 2.5 shows how we embed messages inside a video sequence. For each frame, the message structure is a rectangular M×N grid where each grid block carries one bit of information. We choose a Manchester-like encoding scheme for modulating bits to keep the frequency components of the encoded video signal above 60 Hz. Since Manchester encoding ensures a transition on every bit, it generates less low frequency components in the modulated signal when multiple consecutive bits are identical. For example, for a block with bit 0 encoded, its luminance will be denoted as LOW-HIGH in two consecutive frames. For a block with bit 1 encoded, its luminance will be denoted as HIGH-LOW in two consecutive frames. This modulation signal is then combined with the sequence of carrier image frames. The carrier pixel values inside each HIGH block are increased by α, which means these pixels are made brighter. The pixel values inside each LOW block are decreased by β, which is equivalent to making these pixels darker. In our implementation, the two values are chosen as $\alpha = 2, \beta = 3$. This change is applied to the Y channel in a YUV encoded frame.

2.3.2 Spatial embedding based on texture analysis

The effect of message-hiding to human eyes is not universal across the video sequence. We observe that in some regions, especially in regions having no or little texture, the flicker is more obvious to see. This becomes motivation for us to use texture analysis to select "good" regions to embed in the video sequence. In particular, we seek to categorize the blocks inside each video sequence as "good" or "bad" based on its flicker effect when being encoded by Manchester coding described above. We propose two techniques for this task: one based

on a machine learning technique called **texton analysis** and the other one is **pixel-based texture analysis** method. The former is the more accurate and complete technique for identifying "good" and "bad" blocks, but it is computationally heavy. Therefore, although the technique allows us to explore to what extent of data throughput we can achieve with our texture analysis, for dynamic scene videos, we employ the second simpler method.

Texton analysis. For texture analysis, we employ texture classification based on textons [29,30]. The algorithm is divided into a learning stage and a classification stage. In the learning stage, training blocks are convolved with a filter bank to generate filter responses as shown in Figure 2.6. Exemplar filter responses are chosen as textons via K-means clustering and are collected into a dictionary. After learning a texton dictionary, we model texture as a distribution of textons. Given a block in a video frame, we first convolve it with a filter bank and then label each filter response with the closest texton in the dictionary. The histogram of textons, which is the count of each texton occurring in the labeling, provides us a model corresponding to the training block.

Next, we use K-means clustering to divide our training set of texton histograms into groups. For each group, we segment videos so that only blocks belong to that group are encoded. The videos are then graded based on their level of flicker. Then, each texton histogram group is labeled "good" if the videos have low flicker, and "bad" otherwise. In this manner, we identify the type of texture that is amenable to message embedding.

Each new block of an input video is pre-processed to compare its texton histogram with our training set of texton histograms to find its label ("good" or "bad"). Based on this label, the block is either used for message embedding or not.

Pixel-based texture analysis. Texton analysis is a computationally intensive process and becomes more challenging to use in the dynamic scene videos as the varying content on each frame will require recomputing the "good" blocks to encode. To address this issue, we also propose a computationally efficient method to find the "good" regions to encode. This pixel-based texture analysis is based on the variations of spatial pixel intensities. A larger variation value indicates a more textured region.

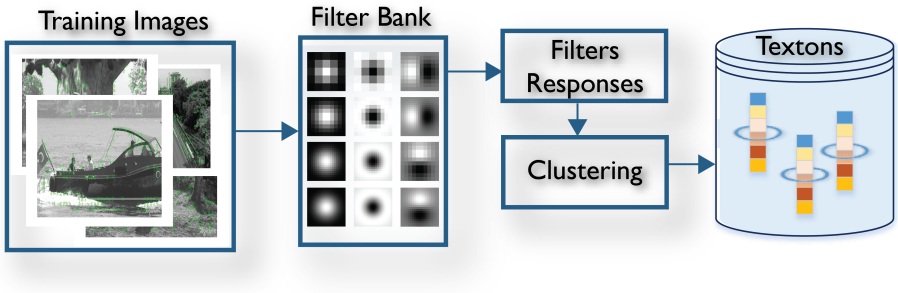


Figure 2.6: Texton Method of Representing Textured Regions

2.3.3 Superpixels

In above sections, we described **the Manchester encoding** at 120 fps, which ensures the frequency component in our temporal video signal are above critical frequency threshold, and **texton analysis**, which excludes the region with the kind of textures that are likely to cause flicker. The flicker, although significantly reduced, is still observable. We observed that the flicker artifacts appear along the edges between checkerboard blocks. This is the result of the phenomenon described in section 2.2, where two neighboring blocks modulated at different phases would cause flicker artifacts at their edges, even at frame rate as high as 120 fps. In addition, these edges are not naturally aligned to existing edges in original video contents, causing visible flicker to human eyes. From these observations, we are motivated to seek another technique to improve the unobtrusiveness of the encoded videos. This technique would ensure: (1) Separate the encoded regions (i.e., get rid of edges between them), and (2) align the border of each encoded region to the existing edges in the original frame.

The technique we chose to fulfill this requirement is **superpixels**, a computer vision technique that provides a convenient primitive from which to compute local image features. It is a method of oversegmentation techniques: an image is divided into sub-regions with respect to image edges, and pixels inside each region are uniform in color and texture. To generate superpixels, we use SLIC (Simple Linear Iterative Clustering) algorithm [31], which is fast and has high segmentation performance.

We use superpixels to determine which pixels inside checkerboard grid to embed information. Recall that in our scheme, pixels inside each block alternate between dark and

bright intensity values. However, we observe that if all pixels inside each block are allowed to alternate, boundary between blocks are perceivable by human eyes. Superpixels, therefore, are employed to limit the region inside each block where pixels are allowed to alternate. This approach also allows boundaries of each encoded region to align with the real edges in the video frame, thus reduce significant perceivable flicker to human eyes.

Although superpixels can align the boundary of each encoded region to the existing edges of the original video frame, the receiver needs to know the location of each super pixel (i.e. which pixels in the original video frame belongs to which superpixel), which means it needs to rebuild the superpixel map for each video frame. The superpixels are also varying in size and shape, which can cause varying quality of decoding. To solve these problems for block-based decoding, we propose a **block-superpixel hybrid encoding method** as follows.

Each video frame is first segmented into superpixels and also divided into checkerboard blocks. In each checkerboard block, we find superpixels that completely fall into that block, and mark pixels inside these superpixels to be encoded. These pixels are then alternated following the previously described method, while other pixels in the block are kept the same. Figure 2.7 shows an example of how pixels inside each block are chosen to be encoded. This hybrid encoding method has the following desirable properties: 1) it aligns the boundary of each encoded region to existing edges in each video frame; 2) it ensures there is no common edge between any two encoded units (blocks), and 3) it allows an easy block-based decoding method—there is no need to rebuild the superpixels map on the receiver side.

2.3.4 Receiver and decoder

The receiver in our system is a camera capable of capturing video at 240 Hz. To evaluate our decoding algorithm, we first captured high frame rate videos from the camera and then extracted all frames inside these videos for offline processing. The offline processing algorithm is implemented in Matlab.

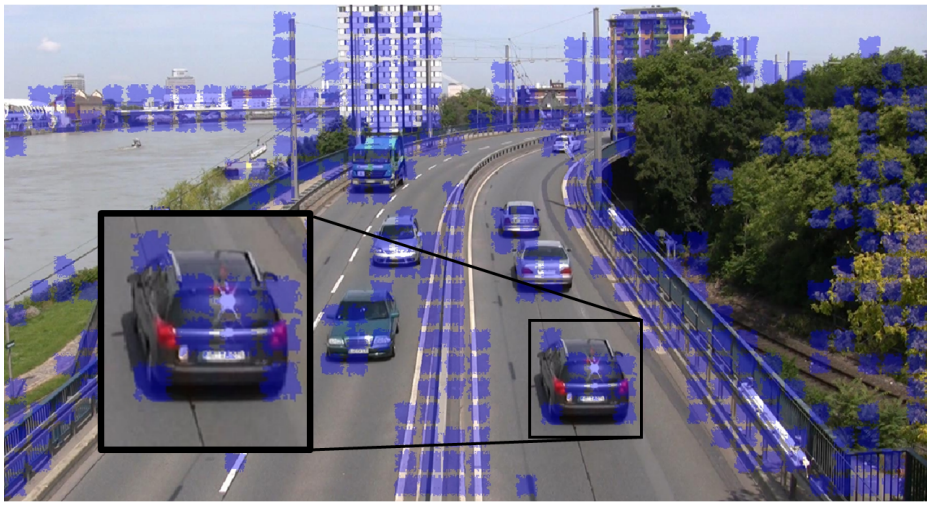


Figure 2.7: Texture encoding method (the pixels inside the blue regions are encoded). The texton analysis avoids encoding in plain texture area in the video frame, such as the road, the sky, etc. while encoding in the high texture area, such as the cars, the buildings, etc. The superpixels method then further separates encoded blocks (avoid boundary effect), and also aligns their boundaries to the existing edges in the video frame.

Frame perspective correction and spatial block division

Because of the capturing angle and the camera distortion, the received video frames are normally trapezoids. This would bring some difficulties to the decoder when recovering the correct location of spatial blocks. Therefore, after extraction, all frames need to be warped into correct perspective. We use projective transformation for frame correction. After a frame has been corrected, it will be divided into blocks for later decoding process.

Decoding algorithm

The main challenge for the decoder is to extract the desired intensity change among the intensity changes due to noise and the video content itself. To minimize the effect of video contents to the decoder, we choose to decode 8 encoded frames from an original video frame. We are able to choose these 8 frames thanks to the following observation: the change by pixel modulation is relatively small compared to the change in video contents. Therefore, by calculating the pixel-by-pixel difference between consecutive frames, we can detect the starting point of each 8-frame group. As we capture the videos at double the screen refresh rate, one frame from the transmitter would produce two frames on the receiver. Therefore,

Algorithm 1 Decoding algorithm

```
Input: a captured video.
Output: a decoded stream of bits.
Extract frames from the capture video;
for each frame in the extracted sequence of frames do
    Warp the frame into the correct perspective;
    Crop the video region inside the frame;
Detect the starting point of each 8-frame group;
for every 8-frame group of the same content do
    for each block inside each frame do
        a_1, a_2,..., a_8 := average intensities of all pixels inside this block in these 8 frames;
        for i = 0, 1 do
            if a_{4i+1} + a_{4i+2} < a_{4i+3} + a_{4i+4} then
                outBit = 0;
            else
                outBit = 1;
            Save outBit to the output buffer;
```

within 8 received frames (or 4 sent frames), each checkerboard block will contain two bits, as can be seen in Figure 2.5. We compare the average intensity of each block over two frames with the average intensity over the next two frames to determine the transmitted bit in this block. The pseudocode for our decoding mechanism is described in Algorithm 1.

Since the objective is primarily to evaluate the limits of spatially adaptive embedding, we assume that the decoder knows the checkerboard size, the original video resolution and the encoded regions for each frame. This eliminates pixel offsets and error for texture analysis on the receiver side introduced from several factors, including video distortion, ambient light change and camera exposure setting. In a full protocol design, these parameters can be included in packet headers or inferred through additional receiver processing.

2.4 Implementation

The implementation of TextureCode consists of a transmitter and a receiver component. For the transmitter, we take an original image or video stream and a data bitstream as input, generate an YUV sequence, and use *glvideoplayer* [32] to play the video at 120 fps on a computer screen, whose refresh rate is set to 120 Hz. We choose an uncompressed YUV format to avoid any artifacts caused by video compression schemes. The receiver is a smartphone camera with high frame rate video recording capability. We chose the iPhone6

since it allows 240 fps capture. It captures the video sequence displayed on the screen and detects the message embedded inside the video sequence.

Currently, both the transmitter and the receiver work offline. We use Matlab to multiplex the original video sequence with the data stream to create an encoded version of the video. For the receiver, we use a Matlab script to post-process the video file recorded on the the iPhone.

We implement two algorithms to evaluate TextureCode. One is the refined texton analysis and superpixels based method for finding "good" regions to encode. The other is more computationally efficient, using pixel-based texture analysis to find the "textured pixels" and encode only in those blocks with a high number of "textured pixels". For the latter method, we leave a few pixels unencoded at the block boundaries to avoid flicker. We use the first method for videos with static scene, as the texton analysis and superpixels are better at detecting regions with near-zero flicker perception. We use the lightweight pixel-based texture analysis for dynamic scene videos.

2.5 Evaluation

We experimentally evaluate the effectiveness of spatial-temporal embedding and its orthogonality to different schemes. In particular, we study the communication link performance of the TextureCode approach in terms of goodput and bit error rate and compare it with the existing HiLight [19] and InFrame++ [18] schemes as baselines.

Experiment Settings. We conducted experiments in a well-lit indoor office room environment using a display monitor screen as the transmitter and a smartphone camera as the receiver. We used an ASUS VG248QE 24-inch monitor to display a set of test videos at a rate of 120 Hz[1]. The screen resolution is 1360×760 while video resolution is 1280×720. The displayed videos were recorded as video streams at 240 fps with an iPhone6 using its built-in camera application in the *Slo-Mo* mode. The default distance between the screen and the camera was set to 70 cm, where the screen fills the camera image. The iPhone was mounted on a tripod as shown in Figure 2.8. We selected a set of 10 videos from two

[1]the maximum refresh rate of the monitor is 144 Hz

bigbuckbunny	Bosphorus	football	highway	Jockey

Table 2.1: The screenshots of some test video sequences.

publicly available standard data sets [33], [34]. Table 2.1 shows screenshots of sample test video sequences.

Metrics. The primary metrics for evaluation are bit error rate and goodput. We chose goodput over throughput since the bit error rates can be highly variable for embedded screen-camera communications. We define goodput as follows.

$$\text{Goodput} = \sum_{\text{all frames}} \frac{D}{t} \tag{1}$$

where D is the number of correctly decoded bits and t is the transmission time.

In addition, we also consider the transmit rate to understand the effectiveness of the spatially adaptive embedding approach. Note that the transmit rate in TextureCode is dependent on the content of the carrier frames, because it encodes more bits in image areas that are conducive to embedding. The transmit rate therefore varies in TextureCode, while it remains constant in the baseline schemes.

$$\text{Transmit Rate} = \sum_{i=1}^{N} \frac{B_i \times b \times V}{N \times F} \tag{2}$$

where B_i is the total number of encoded blocks in frame number i, b is the number of bits encoded in each block, V is the video frame rate, N is the total number of frames in the video sequence, and F is the number of frames needed to encode one bit.

Schemes for Comparison. The two existing techniques can be summarized as follows. HiLight [19] utilizes the alpha channel to encode messages into each frame. In each carrier frame, the alpha value is either 0 or $\Delta\alpha$, which is small (about 1-4%). The messages are embedded using Binary Frequency Shift Keying (BFSK), where 6 frames are used to encode bit 0 or 1 by translucency at 20 Hz or 30 Hz respectively. InFrame++ [18] uses Spatial-Temporal complementary frames (STCF) to design frame structures. In InFrame++, a

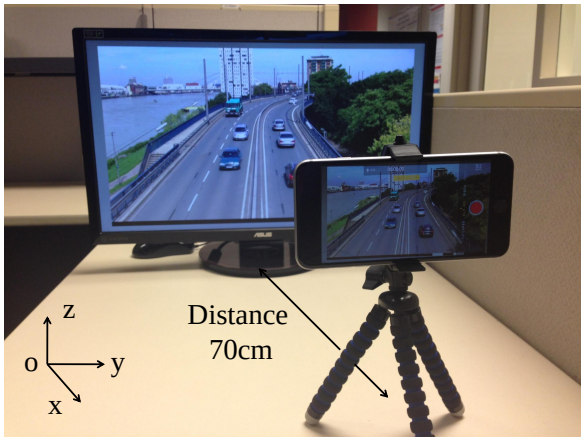


Figure 2.8: Experiment setting

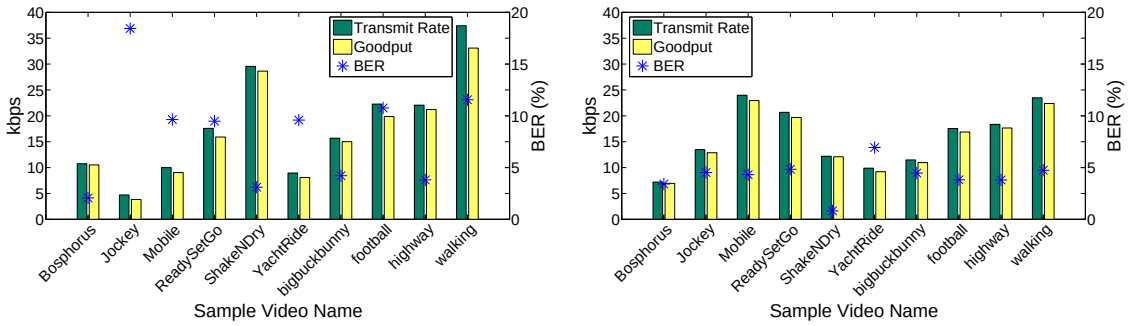
Cell consists of $p \times p$ physical pixels; a *Block* consists of $c \times c$ neighboring Cells, and it is considered the basic information carrying unit. InFrame++'s design boosts data throughput by allowing each block to deliver multiple bits, distinguished by different visual patterns.

For each sample video, we generated one test video sequence each for the three candidate encoding schemes (TextureCode, HiLight and InFrame++) where each image frame of the test video was embedded with a random bit stream. While we used the original implementation of HiLight using the code provided by the authors, we implemented InFrame++ based on the description available in the paper [18], as we did not have access to the code.

In addition, we implemented a *hybrid* encoding scheme where we used our proposed TextureCode technique and the HiLight scheme on different regions of each video frame. As we will show through our evaluations, the hybrid scheme improves the communication performance of HiLight while inducing no flicker.

2.5.1 Communication Performance of TextureCode

We evaluate the communication link performance of TextureCode for two use-cases: (i) *dynamic*, where the visual content (i.e. background) of the test video is changing, and (ii) *static*, where the visual content of the test video does not change. The experimental results are shown in Fig. 2.9, where we plot the transmission rate, goodput, and BER of TextureCode for each test video, for both the dynamic and static cases, respectively. As mentioned earlier, TextureCode achieves near-zero flicker perception for all the tested videos.



(a) (b)

Figure 2.9: Transmit rate and goodput performance. (a) Dynamic scene, (b) Static scene.

The experimental results from these plots indicate that TextureCode has an average goodput of 16.52 kbps for the dynamic case and 15.16 kbps for the static case, while bounding the average BER within 7% for the static case and within 20% for the dynamic case. We observe that the BER achieved in the static case is usually smaller than the dynamic case. This large error spike in the dynamic case happens because the original, unaltered video signal is changing, but our algorithm assumes a constant base-video signal. In fact, the *Jockey* dynamic video sequence has the fastest motion in our test, causing the highest BER (18%).

2.5.2 Comparison of TextureCode with prior work

We compare the performance of TextureCode with HiLight, InFrame++ and the Hybrid schemes in terms of the goodput and flicker perception for the dynamic video cases, as shown in Figure 2.10. We elaborate our inference on each of these dimensions as follows:

Perceived flicker

We observed that while TextureCode, HiLight and Hybrid schemes showed no signs of flicker (flicker level was much below perceivable (subjective) threshold), there was still some residual flicker in InFrame++ at the test viewing distance of 70cm. While there are several proposed objective metrics for video flicker, we are not aware of any metrics applicable to high speed videos. Therefore, the flicker assesment is the subjective assesment of two subjects, according to the grading scale described in Fig. 2.3. It is worth noting that the flicker level of the InFrame++ scheme can be reduced by limiting the encoding block size

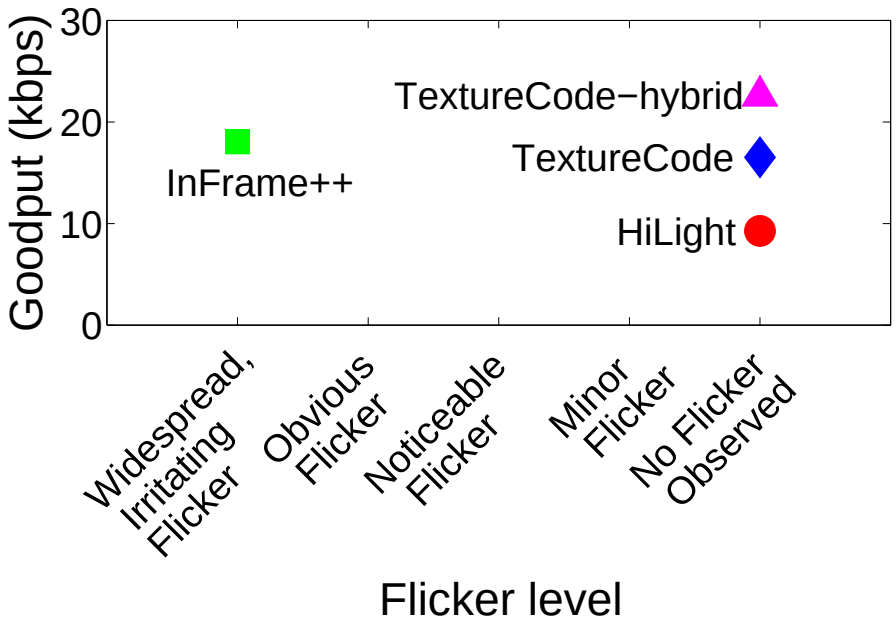


Figure 2.10: Comparison between systems.

	Average (kbps)	**Max (kbps)**	**Min (kbps)**	**Standard deviation (kbps)**
InFrame++ (S)	19.26	23.78	15.93	2.05
InFrame++ (D)	18.05	21.23	15.21	1.63
HiLight (S)	9.66	9.82	9.01	0.13
HiLight (D)	9.27	9.47	8.99	0.1
TextureCode (S)	15.16	22.94	6.96	5.55
TextureCode (D)	16.52	33.08	3.85	9.31
TextureCode-hybrid (S)	21.9	28.68	13.7	4.4
TextureCode-hybrid (D)	22.57	39.13	9.89	8.84

Table 2.2: Summary of goodput for the four systems: InFrame++, HiLight, TextureCode and Hybrid system. **S**: Static scenes, **D**: Dynamic scenes

to 12x12, as described in the the original paper [18]. Reduction in block-size reduces the area over which the block spatial transitions may be perceived by the human-eye. However, a reduction in block size also translates to a reduction in communication range. This is avoided in TextureCode as the design inherently reduces flicker without reducing block size. In particular, the *texton analysis* and *superpixels* methods ensure there are no edges between neighboring encoded units and align the edges of each encoded block to the edges already present in the content of the image.

Goodput and BER

To ensure a fair comparison of the candidate schemes we use a 32×32 block-size for all schemes. We can observe from Figure 2.10 that TextureCode can achieve higher goodput than HiLight, slightly lower goodput than InFrame++, but InFrame++ introduces more visible flicker. The measured BER of TextureCode is 10%, which is also lower than the measured BERs of InFrame++ (31%) and HiLight (40%). It should be noted that InFrame++ improves the goodput by using a smaller block size—a block size of 12×12 would produce a throughput of hundreds of kbps. However, when we experimented InFrame++ with block size 24×24, 12×12 and 8×8, we observed BER close to 50%, which offers virtually no usable capacity. This is the result of the inter-symbol interference and the pixel offset errors. HiLight encodes bits by modulating the *alpha channel* to reduce human flicker perception. Although this technique significantly reduces flicker, with a block size of 32×32, the luminance changes are not easily captured by the camera, resulting in larger bit errors. We observed that the bit error rate is as high as 40% for HiLight, hence reducing effective goodput to about 10kbps.

The main advantage of using TextureCode is that it selectively encodes pixel regions in the frame that have a high signal-to-noise ratio at the receiver thus reducing the number of errors in decoding such pixels, resulting in a high goodput. To further improve the goodput of TextureCode, we also explored a new hybrid technique, named **TextureCode-Hybrid**, where we employ a mix of HiLight and TextureCode in a screen-camera communication system. In particular, we use TextureCode in "good" (high texture) blocks, and apply HiLight encoding to embed messages in the "bad" (plain texture) blocks, resulting in a higher transmit rate. This technique still ensures that there is no flicker in the encoded videos. Table 2.2 shows the average goodput (averaged over all test video samples) of the four candidate schemes: InFrame++, TextureCode, HiLight and the Hybrid systems. On average, the hybrid system achieves 22 kbps of goodput, increasing the goodput of TextureCode by 45% and the goodput of HiLight by 125%. There is a larger deviation in goodput of TextureCode and TextureCode-hybrid systems, because different texture content results in different amounts of encoded video content. It is worth noting that HiLight was

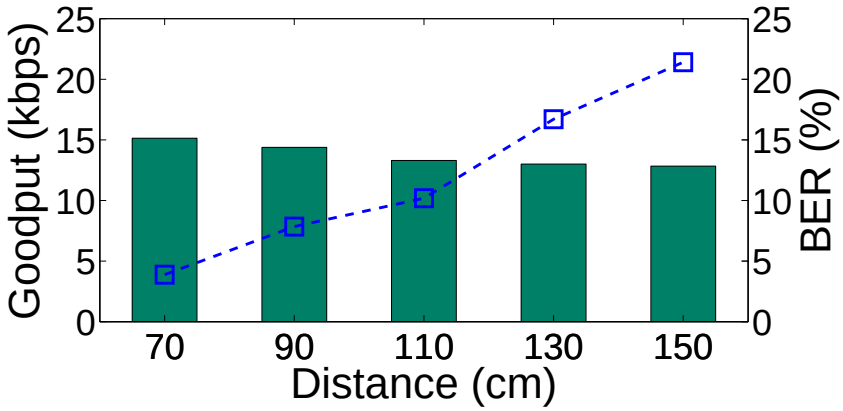


Figure 2.11: BER and goodput vs. distance.

demonstrated in real time and InFrame++ achieved online encoding while the algorithm for TextureCode currently runs offline. We plan to address this in future work.

2.5.3 Microbenchmarking

Communication Range

We examine the communication range of TextureCode by measuring the goodput and BER at increasing screen-camera distance from 70 cm (the minimal distance at which only the screen pixels project onto the camera image) to 150 cm. We plot the average bit error rate (averaged over all videos) and goodput in Figure 2.11. The block size is 32×32 in this experiment. As one can expect, the BER increases with distance as the camera-captured block size becomes smaller, resulting in higher inter-pixel interference [35]. We observe that TextureCode performs well when the distance is within 1 m, maintaining bit error rate less than 10% on average. One possible solution to improve performance at greater distances is to adaptively change encoded block sizes. We reserve such considerations for future work.

Maximum Transmit Rate

Table 2.3 shows the maximum transmit rates for HiLight, InFrame++, and TextureCode. A major improvement through TextureCode is that it can embed a bit for each block within every two frames of the carrier video, while HiLight requires 6 frames and InFrame++

System	Capacity
HiLight	$\frac{frameRate*N}{6}$
InFrame++, $\tau = 4$	$\frac{frameRate*N*bitsPerBlock}{8}$
TextureCode	$\frac{frameRate*N*encodedPercentage}{2}$

Table 2.3: Comparison of Maximum Transmit Rate, which is normalized for the video display frame rate and the number of blocks per video frame. In InFrame++, *bitsPerBlock* is the number of bits per encoded block. In TextureCode, *encodedPercentage* is the percentage of encoded regions over the whole video frame.

requires 8 frames (including about 4 transitional frames) respectively. Although in TextureCode, the maximum transmit rate is reduced by a factor of *encodedPercentage*, we observed that this factor is about 30-40% from our experiments. As a result, the overall theoretical limit on transmit rate of TextureCode is almost of the order of HiLight and InFrame++.

2.6 Related Work

Screen-camera communication. Screen-camera communication began with codes that are not embedded. PixNet [36] uses 2D OFDM to modulate high-throughput 2D barcode frame, and optimizes high-capacity LCD-camera communication. COBRA [37] is a color barcode system for real-time phone to phone transmission optimized for reducing decoding errors caused by motion blur. Another orthogonal class of work includes resolving the frame synchronization problem [38], extending the operational range [39], boosting the reliability and throughput of the screen-camera communication link [40]. More recent studies have focused on embedded screen-to-camera communications. Visual MIMO [41] is a real-time dynamic and invisible message transmission between screen and camera. VR Codes [42] is an invisible code to human eye, which uses high-frequency red and green light to transmit data to a smartphone's camera, where only the mixed colors are perceived by human eyes. HiLight [19] leverages pixel transluency channel to encode data into any screen. InFrame++ [18] uses complimentary frame composition, hierarchical frame structure and CDMA-like modulation to embed messages into videos. TextureCode differs in that it explores spatial coding, an orthogonal dimension to these previous works.

Video watermarking and steganography. Video watermarking and steganography

also make the embedding into images and videos imperceptible [20] [21]. However, the techniques do not address real-world challenges in screen-to-camera communication channel as our system does. We made observations as to finding appropriate regions (in an image) for embedding inspired by the work in watermarking community and proposed a novel technique that addresses screen–camera channel distortions by encoding over spatio-temporal dimensions.

2.7 Conclusion and Future Work

In this work, we study high-rate flicker-free embedded screen-camera communication. An examination of factors that affect flicker perception leads us to explore the spatial dimension of the design space and to combine it with more conventional temporal schemes. The resulting encoding scheme, TextureCode, is spatially adaptive based on texton and superpixel analysis. Experimental results show that this approach reduces flicker to unobservable levels while offering the potential to meet or exceed the goodput of existing schemes. Realizing this potential will still require a receiver that can automatically recognize and adapt to the changing encoding regions in the video stream.

These results also show promise for significantly improving the performance of embedded screen-camera communications through techniques that jointly use multiple dimensions of embedding. This motivates future work to design such protocols, more complete receivers, and online or real-time encoders.

Chapter 3

Body-Guided Communications: A Low-power, Highly-Confined Primitive to Track and Secure Every Touch

3.1 Introduction

As users interact with an increasing number of devices, our interaction times with each device become shorter and the overhead of conventional user identification, authorization, and authentication solutions places an increasing burden on users. Ensuring authorization or accountability is particularly challenging in environments where devices are operated by groups of people. Consider an intensive care unit with multiple patient monitoring and life-support devices, that may be operated while several people including nurses, doctors and patient visitors are present. In some cases, the interaction with a device will only be a single touch before moving on to another device or task. How can we support accountability and auditing by tracking which users looked up information or changed a setting at any given time? If desired, how can we ensure that only authorized users operate these devices? Similarly, challenges arise in numerous other scenarios, from industrial or manufacturing settings to the home environment.

Current approaches broadly fall into the categories of passwords, biometrics, and tokens with short-range radio or near-field communications (NFC). Passwords are cumbersome to use for one-touch interactions and require a user interface for entry that is not present on all devices (consider Amazon's Dash button [43]). Biometrics can be convenient if directly integrated into the interaction (e.g., a fingerprint sensor in the button) but require a sophisticated sensor that adds cost, particularly if every button on a device should have this functionality. Radio tokens, as in keyless entry systems for cars, are more convenient to use but their signals can be easily intercepted, requiring cryptographic protocols. These

operations consume significant energy and the implementations of these protocols are surprisingly often flawed [44, 45]. They are also difficult to secure against man-in-the-middle attacks [46]. Near-field communications can reduce but not eliminate the probability of adversarial interception. Achieving a higher level of security usually requires near-touch between the token and the receiver, such as holding a watch or phone against a payment terminal or a signet ring against a tablet screen [47]. This is an extra step that a user needs to perform, which adds inconvenience. None of these techniques can, therefore, provide a convenient and low-complexity solution to securing quick touch interactions on small devices.

This work explores *body-guided communications* as a primitive for tracking and securing every touch. This allows a wearable touch token to exchange credentials with a receiver through a low-power communication channel that is established at the time the user touches the device. While our technique builds on prior research on touch and body communication [48–52], it differs in that it seeks to create a highly-confined, low-power communication channel between the user's token and devices that is suitable for touches. More specifically, it aims to maintain data rates suitable for touch authentication while improving security by *confining the signal to a few centimeters around the hand and lower arm carrying the transmitter token.* Therefore, we refer to this technique as body-guided communications rather than body communications.

The body-guided communications technique is motivated by an intuition that wearable devices such as a wristband or a ring are particularly suited as security tokens since there is less chance that a user will misplace them and that such devices are in close contact with the body. We also interact with many smart devices through touch, meaning that the human body creates a temporary connection between the device and the user's wearable. This intuition leads to the following fundamental questions. First, can the human body provide a robust transmission medium for body-guided communications in a variety of typical device touch scenarios? Second, can such body-guided communication achieve security properties more akin to those of a wire but with the convenience of wireless communications? Further, can it allow low-power communication at data rates fast enough to execute security protocols during the time of a quick touch?

In this chapter, we introduce a body-guided communications model, touch token design, and a prototype for body-guided touch communications. Body-guided communications require closing the circuit through a capacitive return path which is dependent on exact token positions, posture, and environmental factors. To examine the feasibility under different conditions, we prototype two form factors, a wristband and a ring, and study the robustness of touch communication in several touch scenarios such as a button-device, and a handheld smartphone.

While strong cryptographic security protocols can also be implemented with such a device, the current prototype concentrates on exploring the body-guided communication primitive and demonstrates feasibility with a basic passcode protocol, where the wristband stores and transmits a code to identify and authenticate a user. When the user touches an object equipped with a touch receiver, such as on tablets or medical devices, this identification will be transmitted through body-guided communications to the touch receiver and authenticates the user. The current prototype's data rate is about 1kbps, sufficient to transmit a secret key of length 128-bit on most touches longer than 200ms. Higher data rates are also possible.

We show through experiments with this prototype that by including the human body in the communication channel, the human finger effectively "extends" the transmitting electrode to be very close to the receiver, therefore allowing very low power at the transmitter side. This improves communication energy-efficiency but also protects against eavesdropping and man-in-the-middle attacks on this channel. In particular, we also show that in other directions in which free air has very high impedance, an electrode needs to be within centimeters of the transmitter to eavesdrop on the transmitted signal.

In summary, the salient contributions of this work are:

- Proposing, analyzing and modeling body-guided communications.
- Designing a body-guided low-power authentication token for device interaction through touches.
- Designing an alternative transmitter, that allows reception of signals with unmodified capacitive touchscreen hardware.

- Implementing a prototype and experimentally studying its performance in authenticating every single touch.
- Conducting experiments with these prototypes in three different adversarial scenarios to evaluate the eavesdropping resilience of this design.

3.2 Threat Model and Background

3.2.1 Threat Model

Token-based security protocols rely on detecting the presence of a security token during authentication by exchanging information between the token and the authenticating device. We consider an adversary that seeks to eavesdrop the transmitted signal, either to capture a secret passcode or as a means to launch man-in-the-middle relay attacks (e.g., [46]) on more secure one-time passcode protocols.

We assume that the adversary can design a custom receiver to accomplish this, and that this receiver can be more capable than the receivers used in the wearable and small IoT devices that the user may touch. For example, in the case of the radio frequency signals, the adversary could use a high-gain directional antenna and low-noise receiver to capture weak signals. Similarly, for magnetic coupling-based communications, a larger coil with an iron core would be able to increase signal received at the adversary position. Both of the above devices are simple and can be easily hidden from users. In this work we do not focus on attacks on the wearable or the touched device itself.

3.2.2 Existing Wireless Technologies

We categorize existing wireless methods for communicating with security tokens based on the following three criteria. We focus here on *physical layer* properties since upper layer cryptographic methods are equally applicable across all these technologies yet do not solve all security issues. For example, man-in-the-middle attacks are usually still possible, thus improving physical security is still desirable.

- *attack window*: considers the range from which the adversary can intercept or inject

Communication method	Attack window	Power	Touch association
RF	High	Low ($\approx$ nJ/bit)	No
Magnetic Coupling	Medium	Low ($\approx$ nJ/bit)	No
Vibration	Medium	High ($\approx 100\mu$J/bit)	Yes

Table 3.1: Comparison of existing communication methods.

signals as well as the availability of known techniques to increase this range.

- *low power*: power consumed in the wearable token should be low.
- *touch association*: the ability to associate every touch with the intended signal.

Table 3.1 presents a summary comparison of the communication methods across these criteria.

Radio-frequency communications. Data is modulated on a high-frequency signal with a wavelength short enough so that it launches a radiated wave from the transmitter antenna. Transmitter antennas frequently use an omnidirectional pattern, where signal power is distributed evenly across all directions. In this case, the signal is not confined to the intended receiver. A nearby eavesdropper could receive equal or even stronger signals, resulting in a high attacking window. Simple reducing transmission power also reduces the signal at the intended receiver. Directional antennas are larger in size and a directional transmission may still reflect off other objects in unwanted directions. Security-oriented beamforming and other physical layer security techniques can reduce this attack window [53], but it is difficult to apply such techniques to wearables and small IoT devices for several reasons. First, information about the channel state is often needed in advanced, which is impractical for mobile wearable devices. Second, for directional transmissions or beamforming, the size of an antenna array with a reasonably narrow beam angle would be at least 10 times the wavelength. Since the antenna is constrained by the wearable form factor (ring: about 1-2cm, wristband: 5-10cm), the frequency of the radio would have to be tens of GHz. Operating the token at this frequency range consumes significantly higher power than at lower frequency (100-200KHz), so it is less suitable for a small battery-powered wearable device. More problematic is that the adversary may be less constrained in size and could take full advantage of high gain antennas and sophisticated receivers.

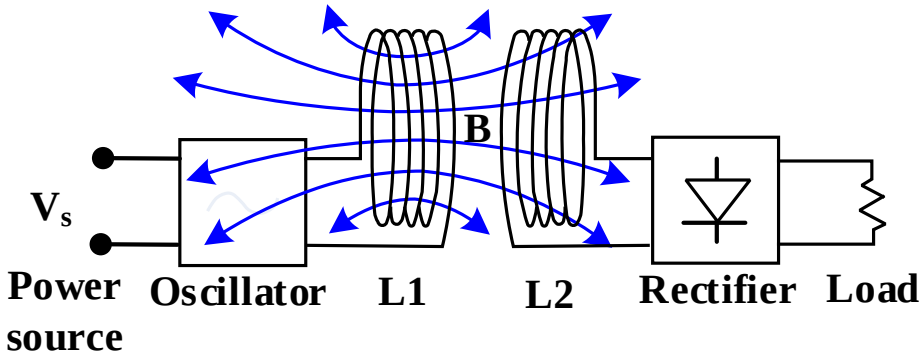


Figure 3.1: Magnetic coupling.

RF communications can be optimized for energy consumption resulting in about 10 to 100nJ/bit for transmission [54, 55]. Since it is difficult to confine a radio wave to a very short distance, the association of a device with a user touch is not clear when multiple users are around.

Near-field communications: Magnetic Coupling. In this technique, power is transferred between coils of wire through a magnetic field. In Fig. 3.1, an AC signal generates an oscillating magnetic field around the transmitter coil L1. the part of the magnetic field that passes through the receiving coil L2, generates a corresponding AC current in the receiver. Magnetic coupling is more limited in distance since the field strength reduces with distance cubed and the fraction of the magnetic flux passing through the receiver coil depends on orientation alignment.

However, an adversary has several options to increase the received power. The adversary could simply use a larger coil with more turns. Further, without space and cost constraints of a small device, the adversary can add an iron core inside the coil loop, since this material has very high permeability (>10000), thus it concentrates the magnetic field towards the adversary [56]. As a result, while more difficult than for radio frequency, any nearby adversary could still achieve higher signal-to-noise ratio than an intended receiver. As an example of attack risks to magnetic coupling-based communications, although NFC has a nominal operating range under 10cm, previous work [57] showed that it is possible to eavesdrop an NFC channel at a distance of 20-90cm, using a loop antenna that couples well with the magnetic field. Therefore, the attack window for magnetic coupling is ranked medium.

The power consumption of magnetic coupling tends to be low (transmission energy $\approx$ nJ/bit [54]), comparable to RF communications. However, since magnetic coupling authenticates all token inside the reception range, it cannot fully associate the touch with the intended signal when two tokens are both in close proximity of the receiver.

Vibration. Recently, vibration-based techniques, such as Ripple II [58] have introduced the ability to associate touch with the intended signal by guiding the acoustic signal through the finger bone. Ripple II uses a vibration motor as the transmitter and a microphone as the receiver. It achieves 7kbps from a ring and 2-3kbps from a watch, so it has the potential to satisfy the rate needed for authenticating every touch. Moreover, Ripple II is able to mitigate the attacks on vibratory sounds, but still an adversary with high-speed camera and line-of-sight to the device may intercept the vibrating signal.

However, current prototypes have high power consumption due to the vibration motor [59]. Current consumption of a typical vibratory motor [60] is up to 90mA at 2V, so the power consumption is nearly 200mW. At 2kbps bitrate (from a watch), the energy per bit is 100μJ/bit.

Goal. Among the three methods mentioned above, vibration is the only method with touch association ability, but it can only be achieved by at least three orders of magnitude more energy per bit than RF or magnetic coupling. Our goal, therefore, is to provide a low attack window and touch association at low power consumption, ideally comparable energy per bit as RF and magnetic coupling.

3.2.3 On-Touch and On-Body Communication

Several earlier projects have introduced the concept of communicating upon touch using different forms of body communication. EM-Comm [49] works in reverse direction: information is encoded in electromagnetic emissions of electronic devices and sensed by a receiver in a wristband when the devices are touched. Security was not a focus of this work and given the magnetic component of this signal, the attack range can be expected to be one meter, similar to that of near-field communications. BodyCom from Microchip [61] ostensibly uses the human body to transmit a signal from an on-body mobile unit to an external base unit upon touch. The design relies on capacitive techniques for detecting touch and works well

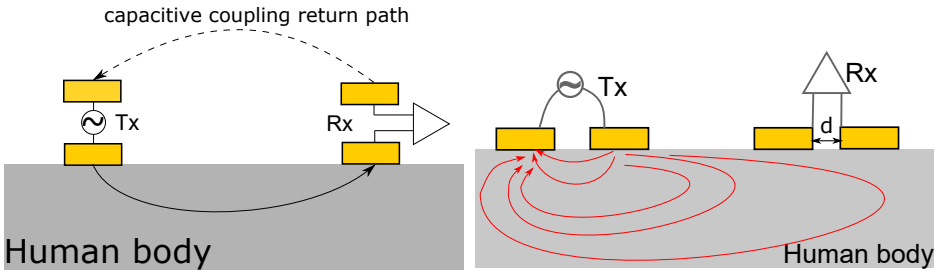
when the user and the touched device can capacitively couple to a large central conductor, such as a door frame or a metal desk, to serve as common ground reference point for both units to close the circuit. The design also includes coils for magnetic coupling, likely to improve data rate particularly when the capacitive coupling is weak. This design also does not confine communications to the human body. Even when only considering the capacitive channel, a significant signal component travels through these external conductors. Moreover, the magnetic component again lends the design similar attack range properties as near-field communication. These techniques, therefore, can provide touch association but do not offer a highly confined attack range.

There are several related works on on-touch communication, which do not focus on confining the signal to a small part of the body. Hessar et al. [48] shows how signals from commodity fingerprint sensors and touchpads can be used to transmit information to other devices in contact with the user's body. Due to commodity device constraints, the data rate is limited to 50bps, which does not allow for exchanging longer codes or executing security protocols in the brief sub-second touch scenarios we consider in this work. Moreover, it demonstrates how the signal can be received anywhere on the human body so that it is available to a broad range of wearable devices. Biometric Touch Sensing [51] also has the same limited bit rate problem: due to the COTS device's update rate, its transmission rate is only 12bps. Our design seeks to satisfy the bit-rate requirement (token is exchanged within one touch) by using a customized receiver that can be easily attached to the current devices. The design also confines the signal more within a small region of the body.

In addition, researchers have explored body communication techniques that can communicate between several devices connected to the human body [50, 62–67]. These also either do not fully confine the signal to a small part of the body or cannot communicate through a finger touch connection. We will discuss these in more detail in the next section.

3.3 Body Guided Communications

To reduce the attack window and power, we seek to guide signals between the wearable and a touched device through the human body.



(a) Capacitive coupling. (b) Resistive coupling.

Figure 3.2: Different coupling types in IBC.

3.3.1 Challenges with employing body communication methods

The goal of transmitting a signal from one body part (at the wearable token position) to another body part (the fingertip) is ostensibly similar to that of intrabody communication (IBC) between two devices coupled to the human body. The challenge with directly employing such body communication methods is that they require direct electrode contact with the human skin for both the transmitting and receiving devices.

Two coupling types are normally used in this communication: capacitive coupling and resistive coupling [66]. In both types, both the transmitter and receiver require two electrodes each. In capacitively coupled IBC (Fig. 3.2(a)), one of the electrodes on the transmitter and receiver side is attached the human body, while the other is floating [68, 69]. In resistive coupled IBC (Fig. 3.2(b)), both of the electrodes in the transmitter and receiver are attached to the human body [65].

Callejon et al. [67] observed that in resistive coupling, the signal attenuation increases with the Tx-Rx distance, while in capacitive coupling the path loss is much more dependent on the surrounding environments since the circuit is capacitively formed through the floating electrodes. In addition, when interelectrode spacing is longer in resistive coupling (either at the transmitter or at the receiver), the signal attenuation is lower. This is because with close spacing, the current mostly flows along the direct path between them. With larger spacing, there exists more dispersion of the lines of current from the direct path, allowing more current to pass by the remote receiver electrodes.

This creates several challenges when applying the above two coupling types to transfer a signal from a wearable token to the fingertip. First, since the fingertip size is small, two electrodes touching the fingertip could only be spaced by a few mm. This significantly

reduces the received power from these two electrodes as we saw above. Second, it is not desirable to require all object touch surfaces to be made of conductive materials (copper, iron, etc.). In most cases, the electrodes could be more easily hidden behind layers of non-conductive materials (plastic, glass, etc.). This means that there is no direct resistive skin contact to the electrode of the touched device and neither the traditional capacitive coupling nor resistive coupling for body communications is possible.

3.3.2 Double capacitively coupled communications

To overcome these challenges with conventional intra-body communications we design a body-guided communications method that allows for a double capacitively coupled circuit.

Design. The key difference in our design compared to previous on-body communications is the combination of resistive coupling at the transmitter side and double capacitively coupling at the touched receiver. As will be seen below, this design improves received signal at the intended receiver while reducing it at an attacker monitoring the channel on air.

On the *touched device*, none of the electrodes have to be in direct skin contact, but one is placed as close as possible to the expected touch-point of the device (usually behind non-conductive material that the device is made of), while the other electrode is simply floating and even less constrained in position. On the *wearable side*, we exploit direct skin contact since this can usually be accomplished for wearables. Both electrodes are placed in direct contact with the user's skin, and their electrode spacing is maximized given the size constraint of the wearable token (wristband or ring).

In other words, the link between the wearable and the user's body is through resistive coupling, while both links between the user's body and the touched device are through capacitive coupling. Note that this differs from conventional capacitively coupled body communications on both sides. The intuition here is that by attaching the wearables second electrode closer to the main body, the large human arm effectively forms a larger capacitor with the floating electrode of the touched device. This creates a stronger signal and compensates for the reduction in signal due to the double capacitive coupling on the touched device while keeping the signal largely confined in the arm.

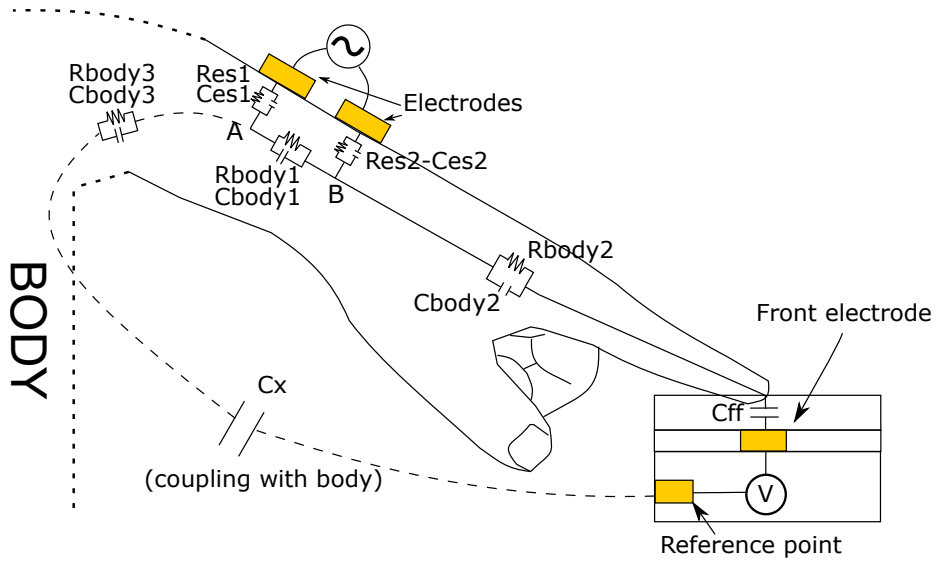


Figure 3.3: Body-guided communication method: Channel modeling.

Our approach differs from Microchip's BodyCom [61] and other capacitive body communication techniques in that the return path directly couples to the body. Thus, it does not require common external ground planes for the two units to couple. This allows the system to work well in more environments and reduces the attack window. Our design also differs from work by Hessar et al. [48]: it allows both electrodes on the touched device to be capacitively coupled, while their work assumes a metal surface with direct resistive skin contact at the receiver side. Capacitive coupling is easier to incorporate into many objects made out of non-conductive materials.

Model. To understand this better, consider the circuit model for body guided communications in Fig. 3.3. The two electrodes in the wearable are powered by an AC signal generator and placed in direct contact with the user's skin. Inside the human body, there are conductive tissues, which are separated from the electrodes by a layer of skin's epidermis. We model the epidermis layer between each electrode and the conductive tissues as a parallel pair of resistor and capacitor ($[R_{es1}, C_{es1}]$ and $[R_{es2}, C_{es2}]$). We separately model the impedance between these 2 points in the conductive tissues under the two electrodes ($[R_{body1}, C_{body1}]$) because the resistance in the tissue is far lower than the skin's. The majority of the current will flow through this skin-tissue-skin path. A second much weaker current path, but one significant for our design, flows through the fingertip and through the

touched device. This path can be modeled as the tissue impedance between point B and the finger ([R_{body2},C_{body2}]) and the double capacitive coupling to the human body. Since the surface of the touched object can be non-conductive, the fingertip and the front electrode forms a capacitor C_{ff}. Finally, the reference point forms a capacitance C_x through the air with the large human body, which is connected through a last impedance with the other wearables electrode A, effectively closing the circuit loop. The voltage at the front electrode is measured by a receiver with respect to the reference point (internal ground) of the device. Note that this ground point can also be a metal surface inside the device.

Note that due to the large distance, C_x is much smaller (pFs) than C_{ff} as well as the tissue or skin impedances (nFs). Therefore, it is the limiting factor on the circuit allowing the signal to flow through the touched device. Since electrode A is also attached to the body, the comparatively large human body can capacitively couple to the device, increasing the capacitance C_x to about 100pF according to the Human Body Model [70].

Consider now the change occurring when the finger stops touching the device. The increasing distance between the fingertip and the front electrode reduces C_{ff}. Since the size of the fingertip and the front electrode are small compared to the size of the human body, C_{ff} becomes smaller than C_x even at very small distances. Then C_{ff} is the limiting factor and the resulting high impedance lets only a negligible current flow through the device. Since the presence of a detectable signal is so closely linked to actual touch, this shows how the finger guides the signal and promises to achieve our goal of touch association and small attack windows.

All other paths through the air have higher impedance than the above path through the body, leading to much weaker signal received at any point on air. For a given double capacitively coupled touch device, we experimented with different setups of the two electrodes at the wearable side: both with direct skin contacts (resistive coupling), one with direct skin contact and one separates from the skin by a thin mylar layer (capacitive coupling), and both capacitive coupling. More details of the form factor of the wristband are in Section 3.4.1. Fig. 3.4 shows the average signal-to-noise ratio at the intended receiver and at a position on air that is 1cm and 5cm away from the token. When the touch device has double capacitively coupled electrodes, the configuration with both resistively coupled electrodes

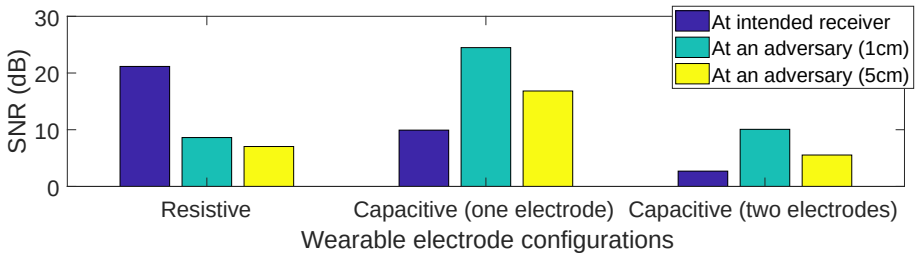


Figure 3.4: SNR at the intended receiver vs. at an adversary on air for different wearable electrode configurations.

on the wearable side gives us the highest signal advantage at the intended receiver over an adversary monitoring the channel on air. This is the rationale for our design choice.

3.4 Touch authentication token design

Let us now consider how to use this body guided communication primitive to design a per-touch authentication token. Our system consists of a transmitter embedded in a wearable token, which is worn on the user's body and sends the user code through the finger to the fingertip. When the user touches an object with an embedded receiver, the receiver can detect the signal and decode the authentication credentials for each touch event. The design sets aside more sophisticated protocols such as time-based one time passwords [71], and focuses on demonstrating the feasibility of improving the token communication with body-guided communications through a passcode exchange from the wearable to the touched device. It assumes that the wearable is activated just before such an exchange.

3.4.1 Wearable Design

Electrode placement and size of the token are key design factors since the body guided communication signal is dependent on body resistance as well as environmental capacitance. The goal is to enable a wide range of possible touch scenarios.

Touch Interaction Scenarios. To guide the design, we chose the following samples of device interaction scenarios: (1) a wall-mounted device touched by a standing user. This represents a switch, smart thermostat, or display for example; (2) a device on a table touched by a sitting user, representing a tablet or touch screen; (3) a user holding a touch device, while touching it with the same hand; and (4) a user holding a touch device, while

touching it with the other hand. In most cases, the actual touch will occur with the index finger of the dominant hand, except for case 3, when touches are performed with the thumb.

Form Factors. Based on the modeling of body guided communications in Section 3.3, we seek to increase signal quality by 1) placing a token close to the intended receiver and 2) maximizing the electrode spacing.

Rings or watch- and wristbands stand out as wearables that fit the distance criterion. Let us, therefore, consider the following electrode designs that maximize electrode spacing within the size constraints of these form factors (Fig. 3.5):

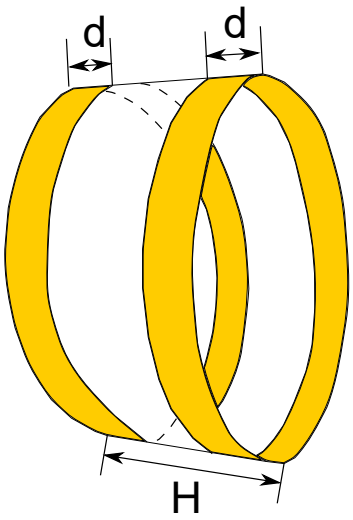


Figure 3.5: Wearable design.

Ring: the ring has the shape of a cylinder with height H = 2cm. There are 2 thin strips of copper on the inner side of the ring (in contact with the finger); they are placed on two sides of the ring and wrapped around the finger. Each electrode strip has height d = 0.3cm, and they are separated by 1.4cm.

Wristband: the wristband has the same shape and electrode placement as the ring, but with H = 2.4cm, d = 0.6cm, and larger electrode spacing of 1.2cm.

Generality of Wristband Design. In order to choose a suitable form factor, in terms of usability and ability to deliver the signal to the intended receiver, let us study the effect of form factor position for the different touch scenarios on the SNR at the intended receiver. For the ring, we then explore two positions: on the index finger, which is also used to touch the receiving device and on the ring finger. For the wristband, we test on both wrists of

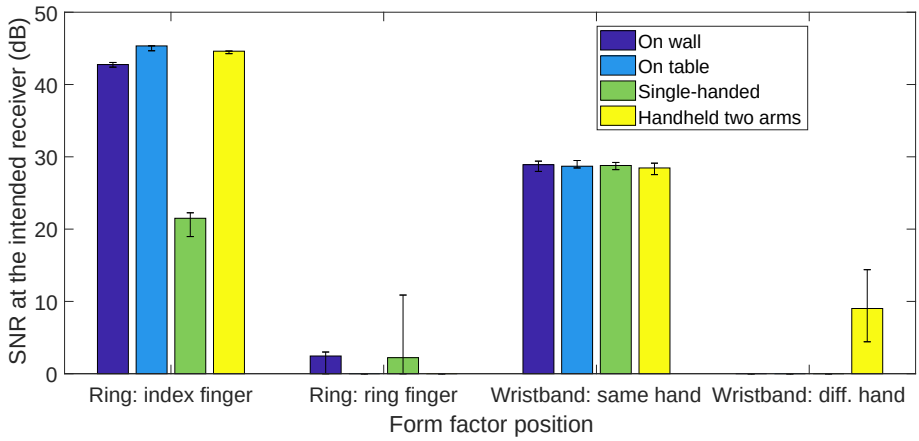
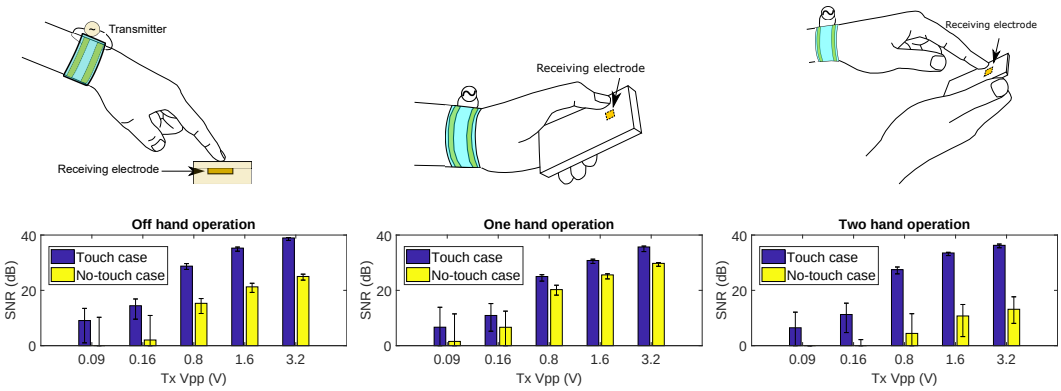


Figure 3.6: SNR received at the receiver for different form factor positions and different touch scenarios.

the hand that is used to touch and on the wrist of the other arm.

Fig. 3.6 shows the signal quality received at the device in terms of signal-to-noise ratio for all combinations of these interaction scenarios and wearable positions. The transmitter is a microcontroller producing a square wave signal at 150KHz, and the receiver has a small electrode pad covered by a thin non-conductive mylar tape. The received signal at 150KHz is measured by a USB oscilloscope that is disconnected from earth ground. We give more details in Section 3.5. As evident, the signal quality varies significantly across these use cases. The index finger ring and wristband form factor provide the most consistent signal quality across all scenarios when the device is located on the same hand, whose index finger touches the device. Since wristbands are more commonly worn than index-finger rings, particularly given the fitness tracker trend, we focus on the wristband design.

We also validate that this form factor achieves our goal of touch association, that is that the received signal is only present when the token-wearing user touches the device. This can be characterized by the SNR difference at the receiver between an actual touch and close centimeter-level proximity. We conduct experiments to investigate this SNR difference for three cases: off-hand table, one-hand, and two-hand operations. We noted that the exact SNR depends on various factors: on the wearable token, the electrode size, the distance between them; on the receiving pad, the electrode size, the distance between the front surface and the electrode, etc. In this specific experiment, the user wears a wristband with dimensions described above, covered by a thin mylar tape layer of 0.1mm. The receiving



(a) Off-hand operation. (b) One-hand operation. (c) Two-hand operation.

Figure 3.7: SNR difference between touch and no touch for different touch interaction scenarios.

pad is a small electrode of size 1cm^2, also covered by a thin mylar tape layer of 0.1mm.

Fig. 3.7 demonstrates the SNR difference between touch and no-touch for three cases: off-hand, one-hand and two-hand operations. The SNR increases with transmitting voltage, but SNR difference between touch and no touch remains relatively fixed in each case. These SNR differences are 13dB, 5dB, and 23dB for off-hand, one-hand and two-hand operations, respectively. As will be shown later, the small SNR difference for the one-hand case would decrease the touch recognition accuracy.

3.4.2 Receiver Design

Since a goal of this work was to provide more flexibility for electrode placement in devices, there are different ways of putting a receiving electrodes into an object that needs authentication/identification. We choose the following example designs:

- **button design:** For small IoT devices like Amazon dash buttons, we embedded an electrode behind its front-facing plastic/glass case. The electrode size is 1cm^2 (about the fingertip size), and the front-facing case is under 1mm thick.

- **phone case design:** For phones and tablets, we can put electrodes in plastic cases used to cover the back of the devices, so that the electrodes have direct contact with the device body. Since the device can be as thick as 1cm, we increase the size of the electrode to be nearly the same size as the device dimension. For example, for a Nexus 5 phone, the electrode size is 13×6cm^2.

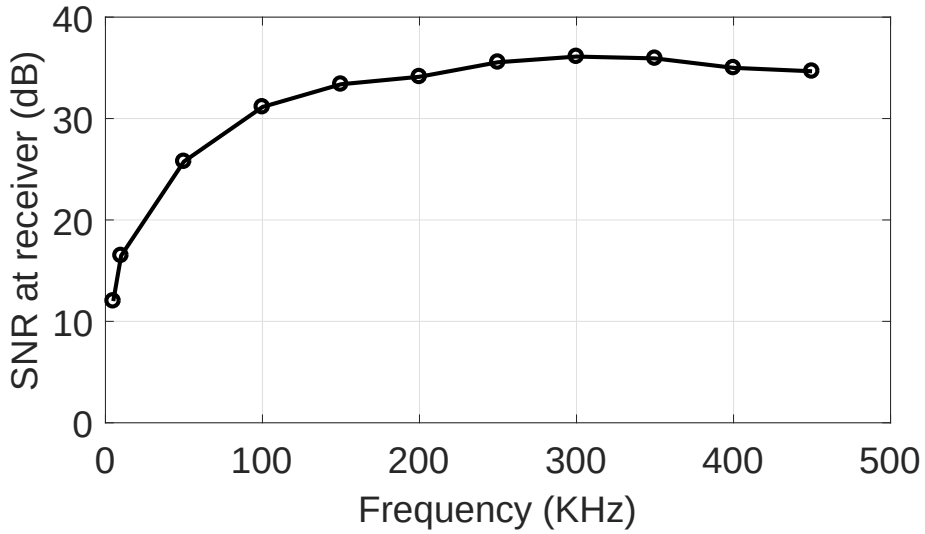


Figure 3.8: SNR received at the receiver for different frequencies.

In these designs, we do not use an explicit second electrode in the device. The receiver connects to the electrode above and measures the voltage with respect to its internal ground.

3.4.3 Transceiver Design

Operating frequency. We look for the optimal carrier frequency for operating the transmitter. Fig. 3.8 shows the SNR received at the receiver for different frequencies when the transmitter sends a 3.3Vpp square wave. Note that the analysis is limited to 450KHz because of the limitation of the microcontroller used for the wearable token. We can see that SNR is worse at frequencies less than 100 KHz, but starting from 100KHz, the SNR doesn't change much with frequencies: the difference is within 5dB. As the result, we should choose frequency above 100KHz to ensure good received signal level at the receiver. On the other hand, the frequency in use should be kept as low as possible since: (i) high frequency means smaller wavelength, but we want the wavelength to be several orders of magnitude larger than the electrode size to minimize any RF radiated signal that an adversary can capture, and (ii) low frequency allows lower power consumption. In all of our evaluations, we choose 150KHz as the operating frequency of the wearable token.

Modulation. The frequency above can be used as the carrier wave for modulating bits in the user's identification code. We choose On-off keying (OOK) modulation method, which represents the bits as the presence or absence of the carrier wave. Given high SNR at the intended receiver when the user touches the device, it is possible to use Amplitude-shift

Keying (ASK) to achieve a higher bit rate. However, we will later show that the simple OOK modulation satisfies the necessary bit rate and code length needed for common per-touch authentication applications.

Authentication process and protocols. For per-touch authentication, the receiver needs to associate each touch with a user ID code. This includes two steps: *touch recognition*, which triggers the authentication process, and *bit decoding*, which demodulates the received signal to get the user's ID code. Touch recognition can be implemented through other components of the device or with the detection mechanism in the signal receiver itself. For packet detection and bit decoding, methods include power-based detection, correlation detection based on known bit sequence (such as Barker sequence [72]). When activated, the transmitter can repeatedly transmit the authentication credentials with a preamble to mark the beginning of a transmission of the code. In this work, we focus on the touch recognition ability of the standalone receiver and a simple power-based bit detection; we leave the design of the full authentication process and protocols for future work.

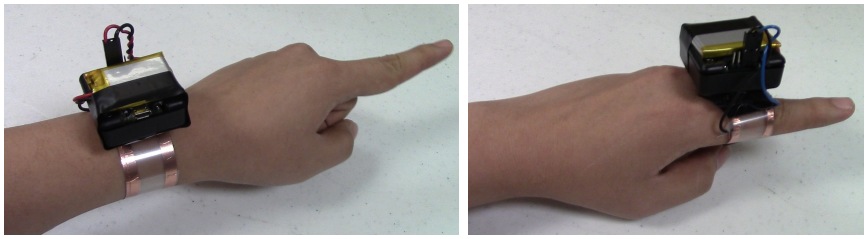
Power. From measurements, we observed that during touch, received signal voltage at the intended receiver is about two order of magnitudes smaller than the original transmitted voltage. For example, when the transmitter is powered by a 3V coin cell battery, the received voltage is about 25mV. We can design a custom receiver to amplify this signal to detect the code being sent; we give details about one such implementation in Section 3.5. For off the shelf phones or tablets, since they are not designed to sense this small signal, we seek a method to generate high voltage at the transmitter to deliver big enough signal to the devices to trigger their touch events.

3.5 System implementation

On the transmitter side, we implement both a low-power token with a custom receiver and a token that allows using off-the-shelf touchscreen hardware as a receiver.

3.5.1 Low power token

Transmitter. We use a Teensy 3.2 board [73], powered by a 3.7V LiPo battery, to generate a square wave of the frequency of 150KHz. This board has a Digital-to-Analog Converter



(a) Wristband form-factor. (b) Ring form-factor.

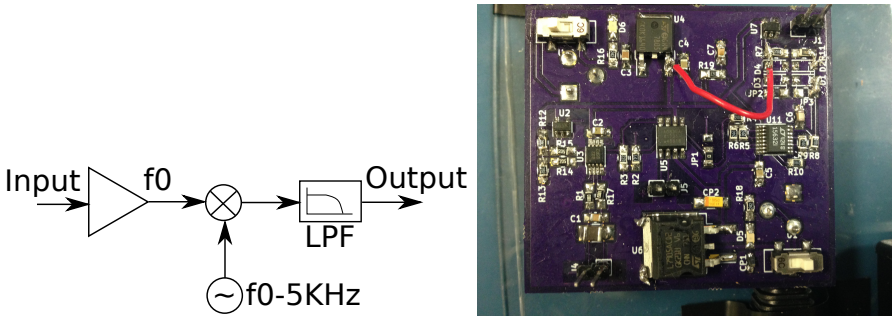
Figure 3.9: Transmitter prototype.

for output voltage control, allowing experimentation with different transmission power levels. The microcontroller output is connected to two electrodes in direct contact with the user's skin. We demonstrate our technique for two form factors of the token: a wristband (Fig. 3.9(a)) and a ring (Fig. 3.9(b)). The microcontroller and battery are inside a small plastic case sitting on top of the electrodes. Note that the electronics of the prototype can be easily miniaturized. The transmitter circuit has much lower complexity than common radio chips and size is primarily determined by electrodes and the battery. It could be integrated into smartwatches as an add-on feature.

Receiver. The receiver downconverts the signal to allow a microcontroller to implement sampling and processing. The design and our fabricated board are shown in Fig. 3.10. The input signal from the sensing electrodes is first amplified with an instrumentation amplifier (INA332 [74]), then fed into an analog multiplier (AD835 [75]) with a reference signal set to $f_0 - 5KHz$, where f_0 is the frequency of the signal generated by the transmitter. The local oscillator is controlled by an Analog Discovery 2 instrumentation device [76]. The output signal from the analog multiplier consists of a 5KHz frequency component together with higher frequency components. By applying a low pass filter (LT1563 [77]) with a cutoff frequency above 5KHz on this output, we can extract the low-frequency component, whose amplitude is proportional to the received signal at frequency f_0.

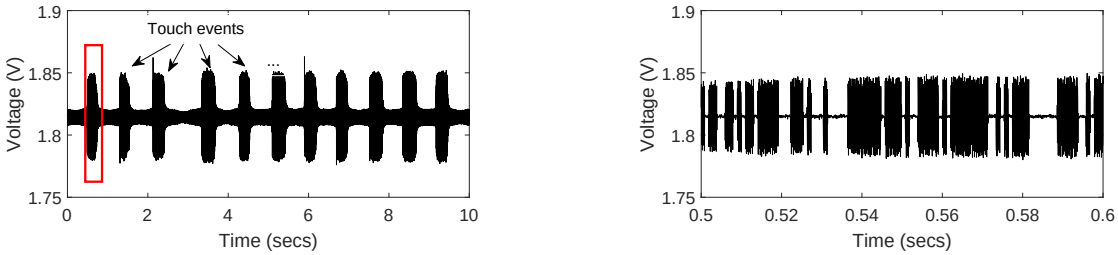
The signal after the low pass filter is read by an MSP432 microcontroller [78] at 20KHz sampling rate. To ensure real-time performance with no sample loss during processing, we implemented a dual-buffered memory, with 2KB for each buffer, to store ADC samples. A ping-pong DMA is implemented so that ADC samples accumulate in one buffer while the processor works on the other buffer.

As an illustration, Fig. 3.11 shows the signal received from the receiver board. The



(a) Receiver design. (b) Fabricated receiver.

Figure 3.10: Touch receiver.



(a) Signal received from the receiver board.

(b) Signal received from the receiver board (zoomed in from red area in Fig. 3.11(a)).

Figure 3.11: Signal received from the receiver board.

user wears the wristband with the transmitter board on the wrist and touches the receiving electrode (for simplicity, the electrode is touched directly here, while the remainder of the evaluation focuses on electrodes that are behind non-conductive material) multiple times with the same hand. The transmitter continuously modulates a random 128-bit identification code on this signal by using On-Off Keying: bit 0 turns off the output and bit 1 turns on the 150kHz signal. As shown in Fig. 3.11(a), the amplitude of the 5kHz signal significantly increases during the time the user touches the receiving electrode and is very weak even when the finger is only about a cm away from the receiver. This helps the receiver recognize touch events and trigger the bit decoding process. Fig. 3.11(b) is the zoomed-in version of one example touch event. At this scale, we can observe the ID code sent from the user token with OOK modulation.

Note that our custom receiver can be easily integrated with smartphones. For the current COTS mobile devices, the receiver can be added in the form of a case with electrodes in contact with the back of the devices and a small receiver circuit inside. The receiver circuit

can send the code received to the mobile device through Bluetooth or USB, and the mobile device can integrate this information with its own touch position identification. For the next generation of mobile devices, the receiver can be made in the form of an ID detection chip alongside the current touch detection circuit and reuse the electrodes in the touch screen as its input.

Our receiver design differs from COTS receivers in the touch sensing mechanism and data rate. COTS touchscreen recognizes touches via the change in capacitance on a matrix of sensing electrodes [79, 80]. It only detects the *presence* and *position* of fingers; its scanning and filtering mechanisms limit the reception of high-speed signals transmitted from the token to the fingertip. In contrast, our receiver is designed to sense the current running through the receiver electrodes when a finger touches the device surface, as described in Sec. 3.3.2. It is optimized to detect signal at the frequency generated at the token transmitter, thus allows much higher data rate, which is needed for per-touch authentication.

3.5.2 Token for COTS touchscreens

In order to elaborate the pervasive of our method to secure every touch with body-guided communication, we show the operation scenario using our custom transmitter along with a COTS touchscreen such as smartphone screen as the receiver. In particular, we generate a modulated signal that will go through the human body and observe the phenomenon at the contact point of user's fingertip and touchscreen. Whenever the modulated signal is transmitted from the signal generator, the touchscreen is affected and *artificial* touch events are generated correspondingly. We confirm that the artificial touches can also be created on COTS devices using the following method, but at a lower rate of communication.

Transmitter. We used Analog Discovery 2 [76] to generate a 10V peak to peak sweeping sinewave signal (200kHz sweep to 500kHz in 1ms) using OOK modulation. The Analog Discovery waveform output is connected to the user's index finger through a wire and ring-like form electrode. The ground pin of the Analog Discovery output is floated.

Receiver. The receiver is a Samsung Galaxy S5 running Android 6.0.1. The app is written on the phone to capture the artificial touch events and decode the transmitted bit sequence using OOK demodulation. Through experiments, we found that the system

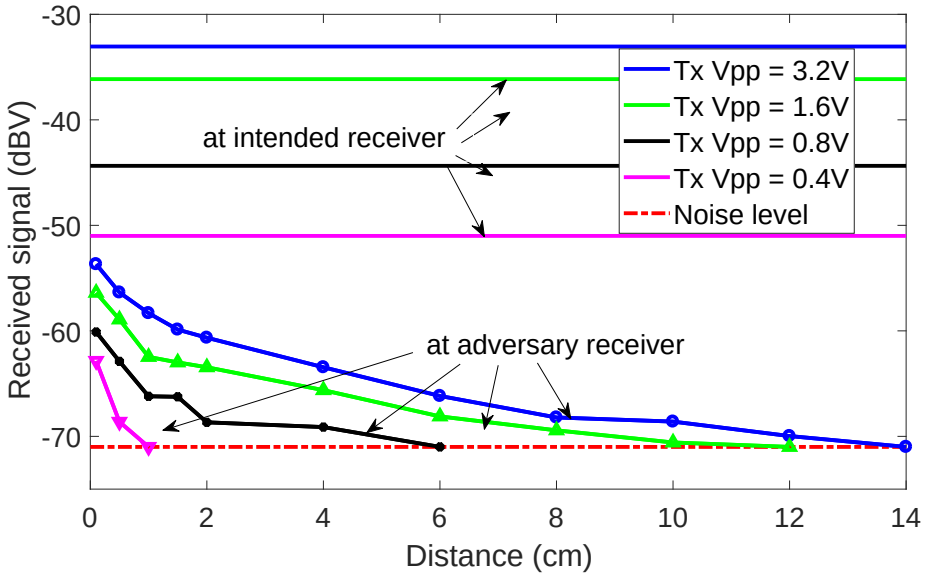


Figure 3.12: Received signal at different distances from the wearable token (wristband form factor).

obtains up to 92.5% of accuracy at 10 bps rate. Details evaluation results are presented in Section 3.6.

We conducted experiments to find out the best waveforms and frequencies that could create reliable communication between our customized transmitter (Analog Discovery) and COTS receiver (Samsung Galaxy S5). We tested the frequencies from 100kHz to 1MHz with sine, square, triangle waveforms. The sine and square waves sometimes can generate expected artificial touches, but we found that sweeping frequency technique obtained better results and is more reliable.

3.6 Performance evaluation

3.6.1 Difficulty of Eavesdropping

Since the received signal at the adversary is dependent on factors such as the transmission power used, we measure the difficulty of eavesdropping as the signal advantage of the receiver, which is independent of transmission power. We define signal advantage as the difference between the SNR at the intended receiver and that at the adversarial receiver. The signal advantage characterizes how easily the token can be designed: a large positive

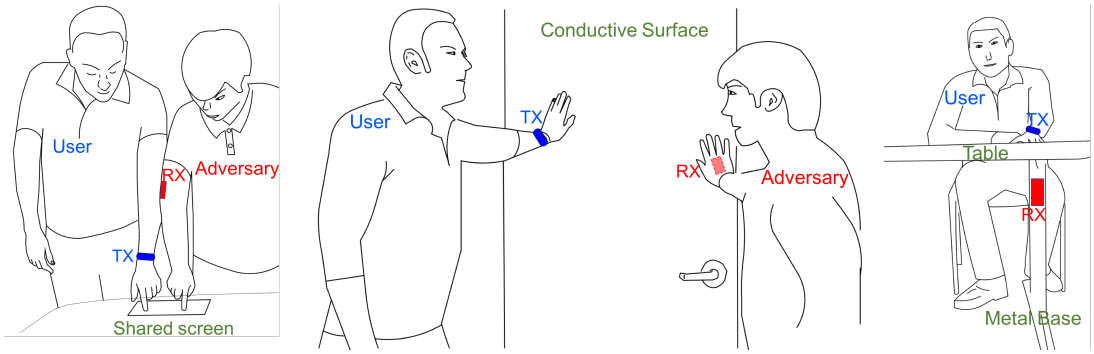
signal advantage allows us to choose an appropriate transmission power to ensure necessary signal level at the intended receiver while reducing the receive signal at the adversary to an undecodable level. A signal advantage equal to or below zero means that this is not possible.

We focus this evaluation on extremely challenging scenarios, where existing wireless technologies cannot achieve positive signal advantages.

Protection against remote monitoring over the air. To evaluate how secure the body-guided communication channel against an adversary monitoring over the air with a wearable-size receiver, for each transmission power, we measure the received signal at a $3{\times}3$ cm^2 electrode over a range of small distances d to the token. We focus on the most challenging case, with very small distances in the mm to cm range. Fig. 3.12 shows the received signal level at the intended receiver and at the adversary, for different distances and different transmission powers. The received signal at the adversary's receiving electrode degrades quickly as distance increases. Even at an extremely close distance of 1mm, the signal received at the adversary's electrode is 20dB worse than at the intended receiver. This means that at our highest transmit power setting the signal was below the noise floor for the adversary at a distance of 15cm. A signal from a well-designed transmitter would be well below the noise floor at mm-range. For comparison, related work [48] reports a signal advantage of 16dB at a distance of 6cm compared to 30dB in our design and requires resistive contacts at both the transmitter and receiver to achieve this.

Note that one cannot expect any signal advantage of the intended receiver with radio or magnetic coupling when the adversary is at such close proximity. As discussed in Section 4.2 the attacker could further take advantage of high gain antennas (for RF) or a larger coil with an iron core (for magnetic coupling), to achieve a strong negative signal advantage, meaning that the adversary has the advantage. These techniques do not apply to body-guided communications.

Low SNR leads to high bit error rate (BER) in the decoding process. Table 3.2 shows the BER using the same receiver for several distances when the transmission voltage is 3.2Vpp. Although BER is 0% when the receiver touches the token, a small gap between the receiver and token increases the BER the BER significantly; at 10cm, the BER is



(a) Adversary touches user's arm.

(b) Indirect touch (conductive material).

(c) Indirect touch (non-conductive surface).

Figure 3.13: Touch-based eavesdropping.

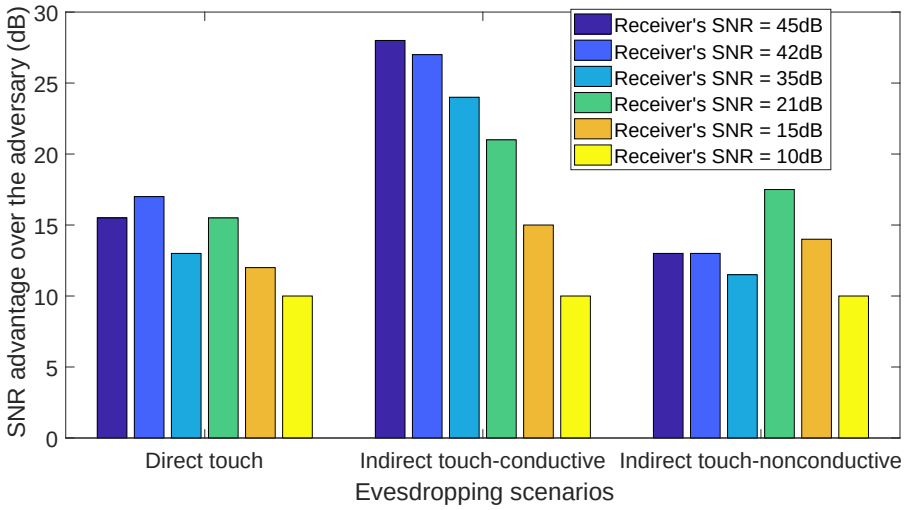


Figure 3.14: Intended receiver's SNR advantage over the adversary.

44.7%, disabling the attacker's ability to eavesdrop the code. This demonstrates how the body-guided communication token design reduces the attack windows.

d (cm)	0	2	4	6	8	10
P(Rx) (dBV)	-53.68	-60.65	-63.45	-66.17	-68.21	-68.60
BER (%)	0	12.78	15.7	28.19	22.7	44.7

Table 3.2: BER vs. distances (received power at each distance is also recorded).

Protection against direct and indirect contact. Besides over the air remote eavesdropping, as can happen in RF security risks, we also consider other example scenarios where an adversary can get in direct or indirect contact with a user to attempt to eavesdrop on his body-guided communications. Fig. 3.13 illustrates these scenarios. To measure the SNR

at the adversarial receiver, we use an Analog Discovery 2 100Msps USB oscilloscope [76] connected with an ungrounded laptop. The noise level is about -71dBV.

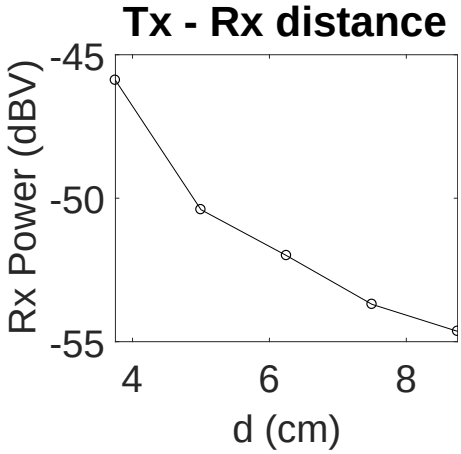


Figure 3.15: Received signal vs. distance on arm.

Scenario 1: Direct touch of user's skin. This scenario represents a crowded or close-collaboration setting where an adversary could achieve direct skin contact without much suspicion while the user authenticates. In this case, the adversary touches the receiver electrode onto the user's skin just below elbow level, as shown in Fig. 3.13(a). For this scenario, the signal advantage remains between 10-16dB across all transmission powers, as shown in Fig. 3.14. We also observed that the received signal power decreases significantly as the receiver moves centimeters away on the arm from the transmitter token (Fig. 3.15). This shows our configuration confines the signal to lower arm carrying the token and virtually no eavesdropping is possible on other body parts.

Scenario 2: Indirect touch through conductive material. This scenario could occur when two persons are both leaning on the metal door, holding handrails in a metro, or on the stairs. In this scenario, we assume that the attacker places his receiving electrode on the hand that touches the metal surface and thereby directly connects to the token user's finger, as shown in Fig. 3.13(b). The intended receiver has an SNR advantage of 21dB over the eavesdropper when the eavesdropper's SNR decreases to 0dB, as shown in Fig. 3.14.

Scenario 3: Indirect touch through non-conductive surface. Here the adversary attaches the receiver to a large metal body hidden behind a non-conductive surface that is touched by the user's hand. An example is the metallic support of a table, as shown in Fig. 3.13(c). The intended receiver has SNR advantage of 10-17dB over the eavesdropper

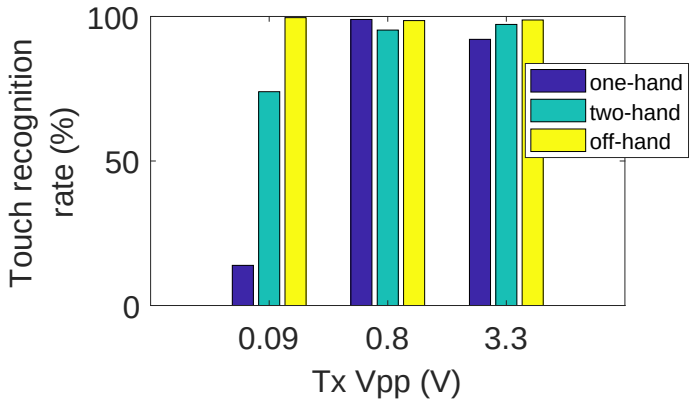


Figure 3.16: Touch recognition rate vs. transmission power.

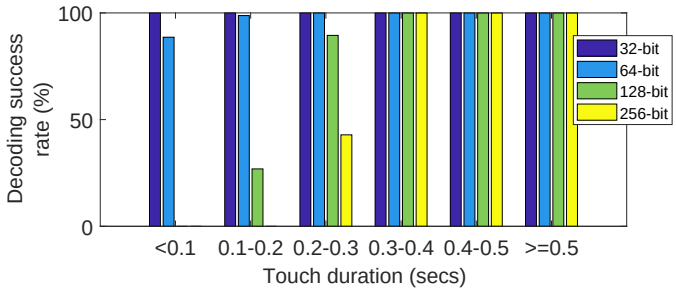


Figure 3.17: Decoding success rate vs. touch duration and code length.

across all transmission powers, as shown in Fig. 3.14.

Overall, these results show that even with direct contact to the user's body the adversary receives a significantly weaker signal than the intended receiver and therefore requires more sophisticated receiver hardware to capture the signal.

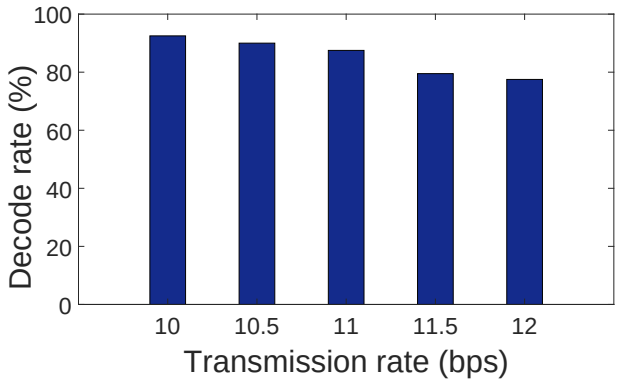


Figure 3.18: Decode rate vs. transmission rate (COTS receiver).

3.6.2 Per-touch authentication/identification

To successfully authenticate every touch, it is important to associate each touch event with one user ID. The receiver should be able to process the signal stream following two steps: (i) recognize touch events, and (ii) detect the user's ID code in the signal portion inside the detected touch event's duration. We evaluate two metrics corresponding to these two steps: *touch recognition rate*, the percentage of the touch events that are recognized, and *decoding success rate*, the percentage of the touch events that the receiver can successfully decode a full ID code that was sent from the wearable token. We also evaluate *bit error rate* of the communication channel for different users. For the following experiments, the users are not constrained on how they touch the device: they can tap or swipe in any direction.

Touch recognition rate vs. transmitted power and touch scenarios. The touch recognition ability can be provided by other components of the device: for example, the Amazon dash button knows when the user presses it, thus can notify our receiver to start decoding the signal. Here we also investigate the capability of a standalone receiver, which can extract touch events from the received signal stream. We tested with 1826 touches for three power levels of the transmitter (peak-to-peak voltages are 0.09V, 0.8V, and 3.3V) and three different touch interaction scenarios as described in Fig. 3.7. A touch event is detected when the amplitude of the received signal crosses an adaptive threshold, which we derive from the statistics of the signal when there is no touch. In our implementation, given S is a window of signal when there is no touch, we choose the threshold to be $T = average(S) + k[max(S) - average(S)]$, and k is empirically chosen to be 1.8. Fig. 3.16 shows touch recognition rate for all these cases. At higher power (0.8V and 3.3V peak-to-peak), the touch recognition rates for all three cases are above 92%. As analyzed in Section 3.4, the SNR difference between touch and no-touch in the one-hand scenario is the lowest, thus at low power (0.09Vpp), the touch recognition rate for this scenario decreases to only 13.81%.

Decoding success rate vs. touch duration and code length. We conducted experiments with two people touching the objects for a total of 2170 touches over 5 days with varying touch durations from 50.7ms to 1.78s. We also experimented with different code lengths: 32, 64, 128, and 256-bit long. The data rate is 1kbps. Fig. 3.17 shows the

decoding success rate versus touch duration. As can be seen, for all code lengths, the decoding success rate increases as the touch duration becomes longer. Also, for the same touch duration, shorter keys have a higher decoding success rate. For the common 128-bit ID, it achieves 89.5% accuracy when the touch duration is between 200ms and 300ms, and 100% accuracy when the touch duration is longer than 300ms.

This result is, of course, dependent on the data rate of 1kbps. The current receiver is limited by the microcontroller sampling rate and not optimized for data rate. According to Shannon theory, the achievable bit rate at 100 kHz is $C = Blog_2(1 + SNR) = 100kHz \times log_2(1 + 100) = 665kbps$.

Bit error rate vs. different users. Since our body-guided communication method relies on human hands as the transmission medium, we examine its performance across different users. Eight graduate students wore the prototype wristband and naturally touched two prototype devices for 5 minutes each: one is an Amazon IoT button [81] with an electrode attached behind its front-facing plastic case, and the other is a Galaxy Nexus 5 phone with an electrode attached on its back. Figure 3.19 shows the bit error rate across these users. As can be seen, for all users and both devices, the BER remains under 10^{-2}. This suggests that with coding robust body-guided communication can be achieved.

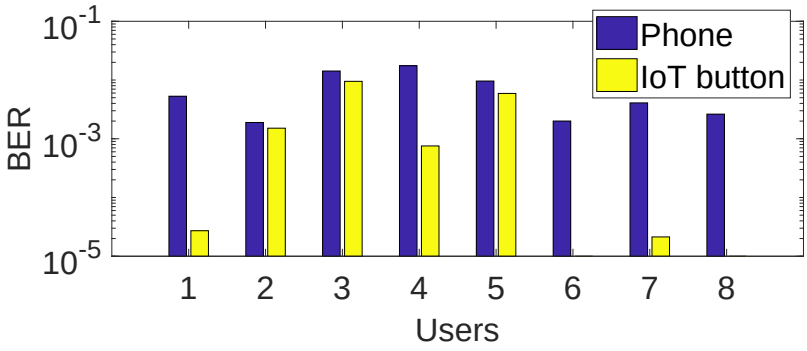


Figure 3.19: BER vs. different users.

COTS touchscreen as receiver. To confirm the feasibility of enabling this channel of communication with an unmodified touchscreen as the receiver, we implemented a simple receiver software to decode the artificial touch event sequence, generated by the Analog Discovery transmitter through the user's body (Sec. 5.2). By counting the number of software-reported touch events during the transmission period (i.e. the effect of the

transmitter to the touchscreen during the period of turning the signal generator on), we achieve a decoding rate of 92.5% at 10bps. When the transmission rate is increased the receiver's performance reduces due to the mismatch between the signal being generated and the response of the screen as shown on Fig. 3.18. While the data rate is low, it can still improve security as part of two-factor authentication protocols, especially over a sequence of touches or during longer swipes. For example, when a user types a password or swipes a secret pattern with his/her finger on the screen, the wearable device can simultaneously transfer a proof that the user possesses the hardware authentication token (e.g., the wristband). In addition, we expect that the data rate can also improve significantly by modifying the touch driver of the COTS receiver for increasing its touch sensing frequency.

3.6.3 Power consumption

The microcontroller in the hardware token only needs to continuously modulate the user code using On-Off Keying, so it can be operated at low power. The results from the prior sections are obtained from our first prototype where the wristband token was implemented using a Teensy microcontroller development board [73]. The average current drawn in this unoptimized prototype is 37mA at 4V supply voltage, which means the token consumes 148mW on average. Given the simple functionality of the token, we started optimizing for power with a low-power microcontroller to understand to what extent the power consumption of the wearable token can be reduced. In particular, we implemented a second prototype token using an MSP430G2553 microcontroller [82] in its low power mode and measured the power consumption of the token when worn on the user's wrist. This prototype is capable of producing the same output signal as the first one, so we do not expect any change in the prior results. Measurement results with this second prototype show that the average current drawn is 1.3mA at the 3V supply voltage, which means the microcontroller only consumes 3.9mW on average. At 1kbps, the energy per bit is 3.9μJ/bit. Even though the microcontroller is not fully optimized yet, the energy per bit is already two orders of magnitudes lower than the estimated power of the only other communication prototype with a smaller attack window (vibration-based communication with 100μJ/bit, see Sec. 4.2).

For comparison, the measured power consumption of our prototype receiver is 525mW.

This consists mostly of heat dissipated at inefficient linear regulators (225mW) and power at the mixer chip (250mW). The power consumption of the receiver can be optimized in an integrated circuit form. Receivers could also be activated by the user's touch to avoid continuous operation but this is out of the scope of this work.

3.7 Discussion and future work

Benefits of body-guided communication over near-field communications. Capacitive coupling is the dual of magnetic coupling: they both occur in near-field region, not in the radiated far field region. However, when the authentication token is worn on user's body, capacitive coupling has an advantage over magnetic coupling: human tissues have a high dielectric constant, so the capacitive coupling approach can alter the electric field to focus on the intended receiver. In contrast, the relative permeability of human tissues is close to that of free-space, so the human body plays no role in guiding the magnetic field. Also, received signals when touch and when no-touch occur (even when the finger is separated only a few mm from the object) have a large difference, which provides a primitive feature for *touch association*.

Security and Activation. Through-body capacitive coupling reduces the attack window by its "beam-forming" ability to create a better channel from the transmitter to receiver than in any other direction. We are not aware of any method that an adversary could employ to increase receiver gain as easily as for magnetic coupling (more turns), RF (high gain antennas), and vibration (high-speed camera). As with wired communications, the adversary can, of course, capture the signal with high quality when directly in the circuit—that is between the finger and the button (e.g., ATM skimming device). Our results also show that the signal can be captured while shaking hands if the signal was inadvertently transmitted during this time. This highlights the needs of one-time password protocols or an activation mechanism (the wearable only transmits when the user touches the intended receiver). The latter would also decrease the token's power consumption.

Currently, our experiments only demonstrate the feasibility of unidirectional communication from the wearable token to the touch receiver. To support sophisticated authentication protocols such as challenge-response, this technique can be complemented with a reverse channel. Note that many protocols can obtain security benefits from our technique even if the reverse channel uses a less secure magnetic or radio-frequency communication medium. For example, the challenge in a challenge-response protocol could be broadcast over Bluetooth or NFC.

Power consumption. The clearly defined channel along the finger also helps lower power at the transmitter, while maintaining a sufficient level at the touched device. Power is also reduced through the operating frequency of hundreds KHz instead of the tens of GHz that would be necessary for RF beamforming approaching a similar level.

There is ample room for optimizing power-consumption of the design. Assuming a highly optimized design with negligible processing power, an estimate for the lower bound can be found in the necessary transmission power. Since the transmitted signal feeds two electrodes in contact with the human skin, two factors affect the transmission power. The first factor is power to charge and discharge the body capacitance: assume the energy per bit is the energy to charge up the capacitance between two electrodes. The measured capacitance is about 10nF, leading to energy per bit at an operating voltage of 3V is $Eb = CV^2 = 10^{-8} \times 3^2$ J/bit = 90nJ/bit. The second part is power dissipated from the body resistance between the two electrodes: The measured resistance is about 10MΩ, leading to power ($P_R = V^2/R$) of about 0.9μW. For 1kbps data rate, the energy per bit consumed by body resistance is 0.9nJ/bit. In total, lower bound of energy per bit of our token is 90nJ/bit, which is comparable to that of common wireless technologies (Wi-Fi, BLE, NFC).

3.8 Related work

Device authentication techniques. Although password, PIN or pattern are widely used for device authentication, they are inconvenient when entering frequently and susceptible to shoulder surfing attacks [83] and smudge attacks [84]. User identification code can also be encoded as a series of electrical pulses that trigger the capacitive touch sensing when the

ring's token directly contacts the mobile's touch surface, e.g., SignetRing [47]. While this ring also allows transmitting a few bits per second when only the finger touched the screen, this rate is insufficient to identify users on a brief half-second touch. Further, since a high voltage is needed to spoof the screen, the ring has high power consumption. Nguyen et al. [85] presented a low-power, battery-free device to transmit data from 3D printed object to the touchscreen. However, the supported bit rate is only up to 32bps, which limits its use in per-touch authentication applications. Also, these approaches still require the tokens to have direct contact with touch surfaces, which is inconvenient for normal touches.

Biometric authentication [86] is another authentication technique used in current devices. Fingerprint identification is currently supported using a dedicated fingerprint scanner, which makes the device design more complex and expensive. Face identification, such as Apple's Face ID [87] identifies the user's face by applying neural networks classifier on images captured by the infrared camera along with the conventional camera. Although our approach also uses dedicated receiver hardware, it offers a different design point. As a much larger number of devices become smart the economics shift so that adding hardware to a few wearables in order to simplify the receiver hardware on each device becomes more efficient. Furthermore, our system allows faster recognition, thus supports authentication on the per-touch basis, not only at the session level as with fingerprint sensors and face identification. Also, the main drawback of biometric authentication is once the user fingerprint/face is captured by an adversary, they are hard to change compared to tokens or passwords. It is also not straightforward to integrate camera-based or face authentication solutions into devices with smaller interfaces or lower specs (such as Amazon buttons), and there is no direct association between people recognized by the camera and actions performed on the touched devices, especially in multi-user operation scenarios.

On-body wireless communication has been proposed for paring wearable devices with smartphones [48]. In this work, they demonstrate transmission bit rate of up to 50bps over the human body using electromagnetic signals, which is insufficient for per-touch authentication.

Per-touch authentication. Different wearable devices were proposed to augment the user's touch with its ID. Bioamp [51] is a wristband augmented with electrodes in contact

with user's skin, and powered by a high-frequency signal source. The signal is then modulated onto the user's body through the skin and transmitted to the user's finger. When the person touches the touch screen, the signal affects the capacitive measurement, and allow the device to decode the modulated information. However, the bit rate is low (up to 12bps), limiting its use for per-touch authentication. IRRing [88] is a ring-like device that continuously transmits the user's ID code in the form of infrared light pulses to a touch device. This helps the touch device associate all touch events inside the region surrounding the point where the infrared light points to. However, this technique still relies on the touch sensing capability of the device for the association, so it cannot be extended to everyday objects. VibRing [89] is also a ring-like device equipped with a vibration motor, which is used to transmit vibration patterns to a touchscreen when the finger wearing the ring is in contact with the touchscreen. Since relying on a mechanical vibrator, the ring can only modulate up to 20Hz frequency, significantly limiting the bit rate of the channel. A vibratory ring is also mentioned as an application of Ripple [58], which claims to be able to achieve 7.41kbps of throughput. However, power consumption was not investigated in the paper.

3.9 Conclusion

In this work, we propose a body-guided communication method for securing every touch interaction from users with a variety of devices and objects. Through prototype touch-token measurements, we showed that the body-guided channel established during every single touch is more secure against eavesdropping than other wireless communication technologies, that is the signal received at the intended receiver is at least 20dB higher than that received at an adversary's receiver in proximity. It can achieve this at low-power consumption of 3.9μJ/bit in an unoptimized prototype, with potential to reach 90nJ/bit. Our current prototype for per-touch authentication is robust enough to reliably deliver a 128-bit ID code on every touch longer than 300ms. We believe this touch token design will provide secure while convenient authentication mechanism for users when interacting with a growing number of devices.

Chapter 4

Light-and-shadow-based Occupancy Estimation and Room Activity Recognition

4.1 Introduction

Building-wide occupancy detection and activity sensing promises to enable a new class of applications across smart homes, elderly care, and retail marketing. In smart homes, for example, it could enhance control of lighting, heating, ventilation, and air conditioning based on sensed and predicted activities across rooms. Useful information ranges from basic occupancy and movement tracking to activity inference (e.g., sleeping, cooking, eating, watching TV or media). In elderly care, activity sensing allows quick detection of emergencies or changes in routine. In stores and showrooms, foot traffic statistics for individual aisles or product display areas are invaluable for ad placement and arranging products.

Existing occupancy sensing technologies. These activities are currently detected by a number of dedicated sensing systems, with Infrared (IR) motion sensing being especially prevalent. Passive or Pyroelectric Infrared (PIR) sensors detect the radiated IR energy from humans and animals [90]. However, PIR sensors require line-of-sight coverage, which increases the number of required sensors to cover a certain area. For example, previous work [91] required one sensor per 4 meter square area. PIR sensors are also sensitive to other heat sources (e.g., hot appliances, sunlight and open window), and they are designed to detect movements, not presence, which limits its tracking of stationary users. For more fine-grained detection in a small area, light barriers detect motion when transmission between an IR transmitter and receiver is obstructed. Other device-free solutions have relied on cameras [92]. Although they are effective and ubiquitous in public places, cameras raise privacy issues, especially in residential areas. More recently, Wifi-based activity sensing (e.g., [11]), has been proposed, which generally achieves large coverage at lower accuracy and faces more

challenges to scale to buildings with many occupants. Besides such device-free sensing, other approaches leverage user devices like smart watches and smart phones (e.g., [93]). The disadvantage of these approaches is that users need to continuously carry, wear, and usually charge them.

More recently, fine-grained localization and activity sensing using visible light has been investigated. Current VLS work mainly uses active techniques (users are required to carry sensors or devices) and focuses on line-of-sight communication between transmitter and receiver [94, 95]. Among passive (device-free) techniques, LiSense [96] demonstrates fine-grained gesture and human skeleton reconstruction using visible light sensing but requires deploying photodiodes on the floor to obtain line-of-sight links with the transmitters. CeilingSee [97] converts ceiling mounted LED luminaries to act as photosensors, to infer indoor occupancy, but requires dense deployment (1.25m between nearby pair) of LED luminaries because of reduced sensitivity of LEDs acting like photosensors compared to dedicated photosensors. None of these technologies can therefore provide device-free occupancy sensing beyond line of sight, which would enable building scale fine-grained activity sensing with lower deployment overhead (i.e. using fewer sensors).

EyeLight Approach. We introduce EyeLight, a device-free occupancy detection and activity sensing system exploiting opportunistic, indirect light sensing so that it can be integrated in a set of networked LED light bulbs. EyeLight forms a mesh of virtual light barriers among nearby light bulbs to sense human presence as they move across the room. Exploiting light provides attractive properties. Due to its nanometer wavelength it is highly sensitive to small motion and objects when compared to RF waves. Also, unlike most RF techniques, light does not suffer from RF interference and cannot penetrate through walls, which preserves privacy and makes it easier to determine in which room an activity occurred.

Contrary to conventional light barriers, however, no direct line-of-sight is needed—the system exploits opportunistic reflections in the environment (e.g., shadows and reflections off the floor). Indirect tracking of users based on their shadows, enlarges the system's operation range, compared to line-of-sight based solutions like PIRs. This allows covering a space with fewer sensors and provides more freedom in deployment locations, making it easier to reuse infrastructure that already exists (for example, recessed can lighting where

power is available but, due to the recessed location, line-of-sight may not exist to the entire space). Such reuse allows for building-scale motion tracking and activity sensing with little installation overhead (no additional building wiring is needed).

The prototype design makes use of the trend of LED light bulbs increasingly containing electronics and having access to plentiful power. Light bulbs are integrated with photosensors and networked to coordinate signaling and to upload sensor data for processing. We design barrier crossing detection as well as occupancy and activity classification algorithms based on sensed changes in the reflected light levels, for example, due to a shadow. This work significantly extends prior work [98] by 1) using dual purpose signaling light (illumination without causing flicker to the eyes while sending the signature of the node), 2) a room-scale prototype with localization and activity recognition, as well as 3) enhancing sensitivity to operate on different reflective surfaces and longer sensing distances (up to 3 meters).

In summary, the major contributions of this work are as follows:

- exploring the feasibility of creating opportunistic meshes of virtual light barriers between modified light bulbs by exploiting reflections off room surfaces.
- proposing a sensitive photoreceiver design for lamp-based light barriers that can detect light reflected from different room materials, including dark floor carpet.
- designing light-based occupancy tracking and room activity recognition algorithms and exploring their potential when deployed across a room's ceiling lighting system.
- designing and implementing a room-scale prototype system and evaluating EyeLight in terms of localization accuracy, estimating occupancy, and recognizing different room activities based on 28.5 hours of recorded data.

4.2 Background and Related Work

Visible light sensing can be implemented directly in illumination systems. Adoption of LED lighting is growing rapidly [99] due to their 75% lower energy consumption and 25 times longer lifetime than incandescent lighting. LEDs can also be switched faster than

incandescent and fluorescent light sources, which allows rapid signaling with light sources and enables novel applications [100]. Given the presence of solid state devices and power converting circuits (AC to DC) in LED light bulbs, it has also become easier to integrate additional electronics in such devices, particularly since power is plentiful. To be acceptable, signaling between lights usually has to be impercetible for human observers.

Human light perception. Imperceptible signaling is possible because human eyes respond slower than photodiodes to light changes. The *critical flicker frequency (CFF)* [24], typically 100Hz, defines the frequency beyond which our eyes cannot perceive time-variant light fluctuation and see only its average luminance. This effect is similar to a low pass filter with the CFF as cut-off frequency. While the exact frequency depends on other factors (such as light intensity, color contrasts, etc.,) sufficiently fast signaling can surpass the flicker perception of human eyes, yet still remain detectable by photosensor front-ends.

Our eyes also perceive light intensity logarithmically, instead of relatively linearly like photosensors. Therefore, a small change of light intensity that is perceivable in a dark room can be invisible in a brighter room. A photosensor calibrated for this range of light levels can easily detect such differences, however.

Existing passive sensing techniques. A major approach to occupancy sensing is using RF signal measurements, based on RSSI ([101, 102]) or time-of-flight [103]. Cameras are also used for monitoring people indoors, but they raise privacy concerns [104]. Other approaches, including Capacitance [105] and Pressure [106] require sensors on the floor, which is not practical for installation in several cases.

Light, both visible and infrared, has long been used for motion detection. Light barriers or curtains [107, 108], for example, detect when a light beam between a source and a photosensor is blocked by a moving object. Since light beams can be easily focused through lenses, they allow more precise movement detection than radiofrequency sensing. To ease deployment, retro-reflective sensors package the light source and sensor into a single device but this usually requires a retroreflector that is carefully aligned to reflect the light back to the sensor.

Visible Light ([85, 94]), an emerging short range communication technology, has been recently explored for indoor localization applications, thanks to the growing use of LED

bulbs. More recent works [96, 97] explored the use of ceiling lights in the visible light spectrum to track people indoor. However, either the photosensors are deployed on the floor to achieve line-of-sight to the ceiling lights, which significantly complicates the deployment, or the LEDs are forward biased to function as light sensors, which leads to lower sensitivity and small coverage in the line-of-sight area.

Challenges in reflective light sensing. Is it possible to achieve both large coverage and ease of deployment by forming a mesh of opportunistic reflective light barriers?. Allowing for indirect, reflective light sensing could extend the sensing range, since movement can be detected not only directly in line-of-sight of a sensor but also anywhere along the longer reflected path of a light signal. Eliminating the line-of-sight constraint also provides more freedom in placing the lights and sensor. In particular, this approach would allow integrating all necessary components into light bulbs, which would significantly simplify the deployment process: the system could be installed by simply changing light bulbs. Note also, that power requirements of the added electronics are met by the power source to the LED light and does not require any battery or additional wiring.

This approach introduces several challenges, however. *First*, the detector now has to recognize much weaker light levels due to two reasons: 1) received light power decreases proportional to square of the distance and reflected paths tend to be longer (for example, the distance with a floor reflection to an adjacent ceiling light is more than double compared to the distance with photosensors directly on the floor), and 2) most typical room surfaces absorb or diffuse a substantial part of the light (e.g. a dark carpet), thus the incident light power on the photodiode is reduced. *Second*, the reflected paths are less well defined. The exact path depends on the position and shape of objects in the space and it is possible that the light reaches the photosensor along multiple paths (akin to radio multipath effects). Motion tracking, occupancy estimation, and activity detection algorithms have to be robust to such effects. *Third*, the receiver should be able to distinguish light from different sources. In addition, any signaling technique used for this purpose should remain imperceptible and not detract from the illumination function of light bulbs.

A common method for detecting a weak signal is a correlation detector with a known pseudorandom number (PRN) sequence. This effectively spreads the signal bandwidth

leading to a significantly enhanced signal to noise ratio. Applying this to EyeLight is challenging, however. First, achieving high processing gains requires long PRN sequences[1]. Given the limited modulation rate of high power lighting LEDs, these sequences would take seconds to minutes to transmit, which is longer than the duration of human movement events that we seek to detect. Second, transmitting continuous PRN sequences with on-off keying would halve the brightness of the ceiling lights, since one can expect equal number of on and off symbols. Third, as a result of the spectrum spreading property, PRN sequences introduce low frequency components, which increases the chance of flicker for human eyes.

4.3 EyeLight design

EyeLight realizes an opportunistic mesh of reflected light barriers through synchronized signaling from networked transmitters and a pulse-based power measurement technique based on sensitive receiver hardware. It relies on modified LED light bulbs to transmit modulated light and contains sensitive photodetectors to detect light signals. It coordinates signaling among light sources so that a virtual light barrier can be established between nearby pairs of lights without interference from other participating light sources. These light barriers are opportunistic since the light needs to reflect off surfaces in the environment to reach the photodetector on an adjacent light bulb. The key rationale for integrating both signaling and sensing components in light bulbs is that it reduces installation and maintenance costs, as power is already available at the lamps.

It addresses the challenge of invisible modulation of LED light bulbs together with self-interference free detection sensitive enough to measure weak reflections through a synchronous, pulse-based power measurement technique. Bulbs emit a periodic pulse, which is short enough to remain imperceptible, meaning it does not noticeably affect brightness of the light and does not cause flicker. Receivers measure the signal power of the pulse and compare it to the overall light level to track movement and changes in the room.

The light nodes have wireless connectivity to report their measurements to a server, where tracking and activity detection algorithms process the datastream to monitor the

[1]For example, GPS system uses 1023-bit PRN sequence which repeats itself every 1ms.

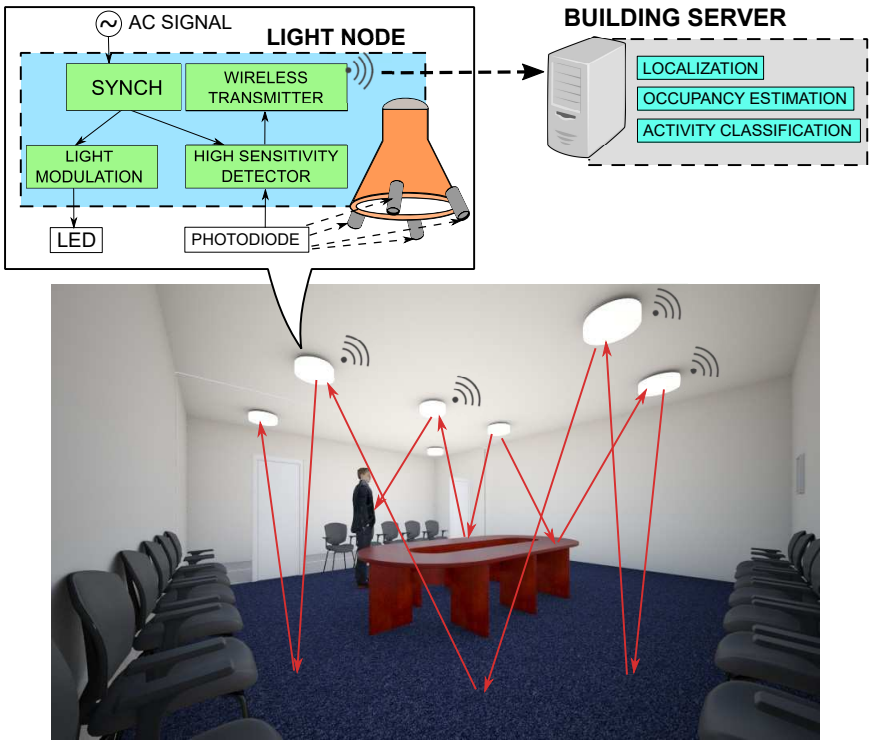
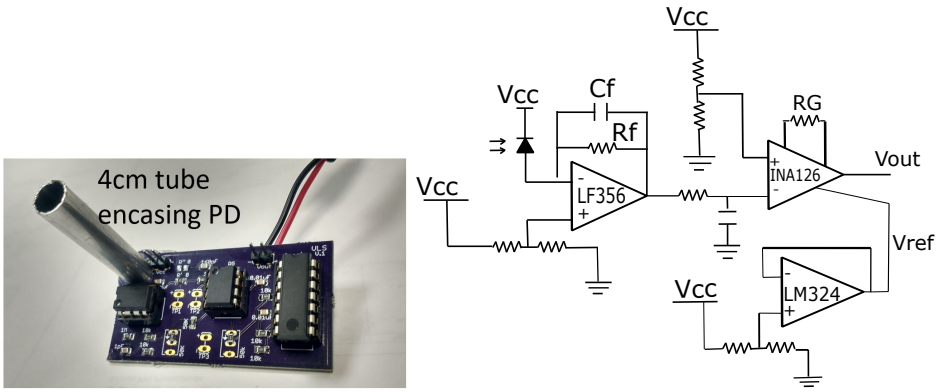


Figure 4.1: Overview diagram of components in EyeLight.

movements and activities of occupants. We assume that light bulbs can be mapped with their location in the room during installation. Self-localization algorithms may also be possible. Fig. 4.1 shows an overview of the components in EyeLight.

Transmitted Signal. The transmitted signal should allow the receiver to separate light emitted by one specific transmitter from other ambient light sources, while remaining imperceptible to the human eye. In theory, this can be achieved with straightforward ON-OFF signaling. Since flicker perception depends on frequency, this raises the question of whether the high power LEDs used in light bulbs can be switched fast enough to remain imperceptible. We measured the rise and fall time of an off-the-shelf LED bulb (Ecosmart 65W BR30) and observed that the lamp takes about 0.1ms to rise to 90% of its peak intensity and a shorter time to fall. This shows that the light bulbs are fast enough for ON-OFF signaling without introducing flicker to human eyes (previous research [24] showed that the critical flicker frequency of human eyes when perceiving a strong single light source is only about 100Hz).

In addition to eliminating flicker, the signal also should not significantly affect the overall



(a) Fabricated Receiver. (b) Receiver Circuit.

Figure 4.2: Receiver.

illumination level. We therefore use periodic signaling, which only occurs in a short slot out of a longer cycle. When ceiling lights are used to illuminate the space, the light would briefly switch off during its slot, while remaining on during the rest of a cycle. This design reduces the lamps' brightness by only a negligible amount. Conversely, when lights are off, the lamps could briefly switch on during their slot to signal. Our implementation focuses on the former. Supporting both modes would require additional calibration of receiver sensitivity.

Receiver. Sensing reflected light off the floor with photosensors deployed on the ceiling is a challenging task. The photsensor frontend needs a high sensitivity to receive weak light and fast response time to detect the modulated signal. These requirements are usually at odds with each other. We achieve these requirements by carefully designing a receiver circuit combining several components (Fig. 4.2(b)). Since we require a fast light sensor to detect short pulse (under 1ms) from the transmitter, we use a photodiode as our sensor. The weak current generated by the photodiode is amplified through a Transimpedance Amplifier. The amplifier acts as the current-to-voltage converter—it converts and amplifies the photocurrent generated by the photodiode to a voltage that can be read out. The amplifying gain of the TIA is set by the feedback resistor R_F following: $V_{out}/I_P = -R_F$.

Compared to a simple detector (a photodiode in series with a resistor R), the transimpedance amplifier has much faster response time than the time constant $R_F * C_d$ (with C_d is the internal capacitor of the photodiode). Therefore, we can use a larger value of R_F

to increase the gain while maintaining fast response at the front end. However, the value of the feedback resistor cannot be arbitrarily large since it is limited by two factors: large Johnson thermal noise ($v_n = \sqrt{4k_BTR}(V)$) can reduce SNR of the frontend, and low input rolloff frequency ($f_{RCin} = \frac{1}{2R_FC_{in}}$) can limit our operating frequency. To further boost the gain, we use a second stage amplifier: an instrumentation amplifier (INA126). The output of the amplifier is given by,

$$V_o = G(V_+ - V_-) + V_{Ref}$$

where V_{Ref} is a reference voltage being fed to the instrumentation amplifier, and G is a controllable gain. One can consider the two inputs to the INA126 as output voltages from two arms of a Wheatstone bridge [98], whose difference we seek to amplify. The negative input V_- is fed with the output of the TIA, while the positive input V_+ is fed with a constant voltage from a voltage divider. Note that G and V_+ are two controllable factors that help the receiver adapt to different light levels.

Multiple Transmitters and Receivers. The previous two sections describe how a single pair of transmitter and receiver can communicate through reflected light on the floor. When multiple transmitters are in the room, each light node needs its own identification—when the sensing module detects a light level change because of a shadow, it needs to recognize which light source created that shadow. Therefore, each LED bulb needs a mechanism to send its own signature. This can be done in the frequency domain, as in [96], or time domain. We choose the time domain because of its simplicity when combined with synchronization from the common AC power signal, which our design assumes. As in other prior work [14], the main idea is that each light fixture chooses its own time slot, during which it signals.

For the time-slot based mechanism to work, the clocks of all light nodes need to be synchronized. We implement this by using the common 60Hz AC signal available from the mains power [109]. Recall that a key motivation for incorporating signaling and sensing into light bulbs was the easy availability of power. We therefore also assume a common AC signal for synchronization. Each zero-crossing event of the mains power signal marks the start of a *cycle* for EyeLight, making the cycle length half the period of the AC signal

(about 8ms).

Given n light bulbs that can potentially observe signals from each other, the system requires n timeslots to uniquely assign a slot to each lamp, which lets the receiver identify the signaling lamp based on the current time. Note that, as in wireless systems, spatial reuse is possible and walls that block light make the reuse of slots across different lamps in a building even easier. This keeps the total number of required time slots relatively small. The maximum number of timeslots that can be supported is determined by the cycle length and the lower slot duration bound derived from the LED rise time.

Besides signaling, each node also looks for signal from other nodes through multiple receivers co-located with the LED lamp. The photosensors point to different directions to detecting signal from surrounding light nodes. For the sampling scheme, we employ a Round-Robin approach to maximize the number of samples per cycle: in each cycle, we let only one photosensor sample the light level in its view, then move to the next photosensors. This ensures each sensor has high enough sampling rate for detecting the fast signal from other nodes.

Fig. 4.3 shows an example of received light power at one receiver over consecutive cycles. This receiver is on node 2, so it observes a big dip in the second timeslot when node 2 signals. It also observes a smaller dip in the first timeslot, when the adjacent node 1 does signaling. This dip shows the effectiveness of our receiver design to sense weak reflected signals off the floor from an adjacent node.

4.4 Tracking Algorithms

The photodiode in each sensor converts the incident radiant energy P to the output photocurrent I_p, making our sensor a light power measurement device. In essence, our tracking algorithms utilize signal power measurements over time, and compare them with the baseline light power level when the room is unoccupied. To improve the confidence of our localization, we introduce two methods, *Spike algorithm* for coarse-grained localization and *Delta algorithm* for fine-grained localization.

The first method measures if there is any change in received light power, which is

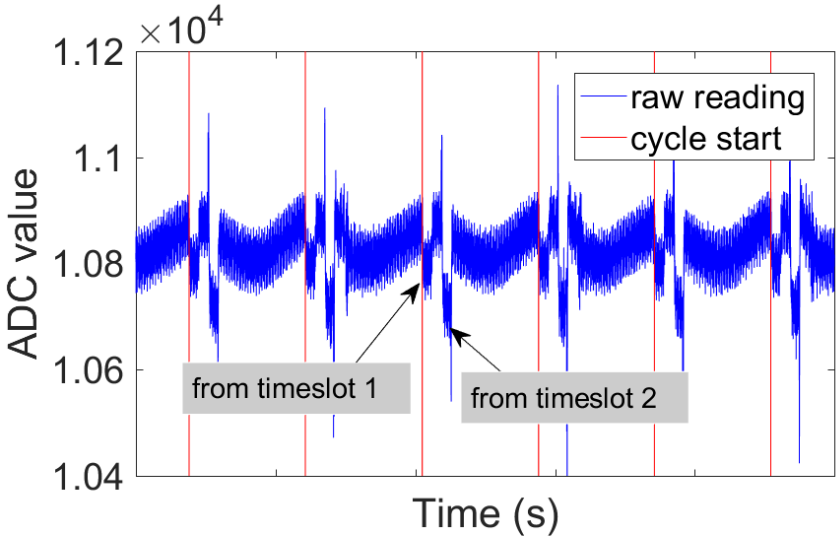


Figure 4.3: Raw readings from one receiver.

caused by movement events surrounding the light node position. We detect this change by continuously taking average received power over an entire cycle for each sensor and using a threshold-based detection to detect when this average power deviates far away from base light level (when the room is empty). This approach, which we call *Spike algorithm*, only tracks movement at a coarse-level—it can only detect if there is a movement event in an area surrounding the spot on the floor a receiver is monitoring.

The second method aims at fine-grained level tracking—it determines whether a change occurred on a specific transmitter-receiver link. With multiple light nodes covering a room and each carrying several receivers, we can effectively create an opportunistic mesh of virtual light barriers to detect when a subject is passing by. Since each light source in the interference domain signals in a unique time slot, receivers can simply check for the presence of the ON-OFF signal in a particular time slot. If the signal can be detected the virtual light barrier is connected, otherwise it is interrupted. This technique is agnostic to most changes in ambient light level that can occur. Over time, the system can then monitor changes in the status of each link.

While the concept is intuitive, its implementation is challenging due to the complex light propagation environment. The system uses reflections off random surfaces rather than direct illumination or a special reflector as in a retro-reflective light barrier. This means

that the light level change when the virtual light barrier is crossed can be small and it tends to differ for every pair of lamps. Moreover, in contrast to conventional light barriers, the illuminating signals are more diffuse and the field of view of the sensor is wider to cover a larger area of interest. In addition, multi-path can exist. This means that signals are often only partially blocked when the barrier is crossed.

To address this challenge, EyeLight employs a delta technique. For a given transmitter-receiver link, it measures the delta change in received signal power when the ON-OFF transition occurs and compares it with a delta obtained under reference conditions (i.e., an occupied room). The signal power delta effectively captures how much light from the signaling transmitter is reaching the sensor. It subtracts out all light from other sources, assuming it remains constant over the duration of one slot. If the measured delta significantly deviates from the reference delta, it means that a change between the transmitter and receiver has occurred.

More precisely, let $P_{i,ON}^{j_k}$ and $P_{i,OFF}^{j_k}$ denote the mean power measured by the k-th sensor on node j while node i is in the ON and OFF phase of its signaling, repectively. We define the delta as $\Delta_i^{j_k} = P_{i,ON}^{j_k} - P_{i,OFF}^{j_k}$.

Note that both the terms effectively sum all light power reached at the sensor k, including both the power from ambient light (natural light and illumination from lamps other than i) and signaling light power received from lamp i. That is $P_i^{j_k} = P_{ambient}^{j_k} + P_{i,received}^{j_k}$. During OFF phase of lamp i, $P_{i,received}^{j_k}$ becomes zero and assuming no change in ambient lighting between ON and OFF phases, it follows that $\Delta_i^{j_k} = P_{i,received,ON}^{j_k}$. This means *the delta value is effectively the light power reflected from node i to sensor j_k during the ON phase of node i*. When a person crosses the link between node i and j, the person can either block light or reflect more light from node i to receiver j_k, depending on the exact position and the reflectivity of the person's hair, skin and clothes. In either case, that causes $P_{i,received,ON}^{j_k}$, and in effect $\Delta_i^{j_k}$, to deviate from the normal level.

This observation becomes the key for our light barrier crossing detection method called *Delta algorithm* (Algorithm 2). Going back to the example of receiver j_k, in each cycle, we calculate the term $\Delta_i^{j_k}$ as described above, then check if this term exceeds a preset threshold range. To reduce noise on the series of calculated delta values, we first apply Hampel filtering

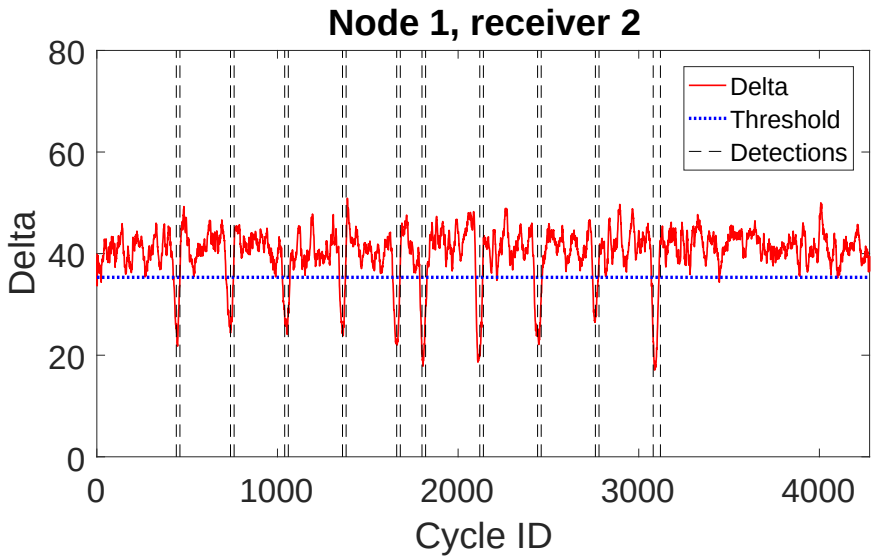


Figure 4.4: Virtual light barrier crossing detection.

to remove outliers and then a low pass filter to smooth the signal. The algorithm then uses a windowing approach (set to 1s) and outputs a detection when the majority of delta values in the window exceed the threshold. We set the threshold based on the mean and the standard deviation of the delta values in the baseline dataset (when the room is unoccupied). (For our prototype, we empirically choose threshold to be $baseDelta \pm 2 * baseStd$). Fig. 4.4 illustrates one output example of the delta detection algorithm, where receiver 2 on node 1 points to node 2's direction, and a person passes 10 times the light barrier between node 1 and 2.

Given detections from either the Spike algorithm and Delta algorithm, we seek to infer the location of the person. For *Spike algorithm*, based on detections of a user or her shadow in the field of view of different receivers, EyeLight derives the user location based on the positions that these receivers are pointing to. We assign a weight for each receiver based on the magnitude of the deviation of the received light power from the baseline level. The final location of the user is estimated as the weighted average of the locations to which the receivers are pointing to. For the *Delta algorithm*, we estimate the location of the user to be the center point between the transmitter and the location the receiver is pointing to.

Note that *Spike algorithm* and *Delta algorithm* compliment each other. The Spike algorithm provides better coverage (any movement in an area surrounding the receiver

Algorithm 2 Delta algorithm - light barrier crossing detection

Input: *readings* from node j_k, $baseDeltas$, $baseSTD$
Output: *events*
while *next cycle exists* **do**
  $cycle$ = getNextCycle()
  for $i = 1 \rightarrow numOfNodes$ **do**
    $P_{i,ON}^{j_k}$ = mean(*cycle period during ON phase of node i*)
    $P_{i,OFF}^{j_k}$ = mean(*cycle period during OFF phase of node i*)
    $\Delta_i^{j_k} = P_{i,ON}^{j_k} - P_{i,OFF}^{j_k}$
    Update $W_i^{j_k}$ *- running series of* $\Delta_i^{j_k}$
    $hampelfilter(W_i^{j_k})$
    $lowpassfilter(W_i^{j_k})$
    if $|\Delta_i^{j_k} - baseDeltas_i^{j_k}| > 2 * baseSTD_i^{j_k}$ **then**
      increase count($events_i^{j_k}$)
    if *end of 1-sec window* **then**
      if (count($events_i^{j_k}$) > window / 2) **then**
        $detection_i^{j_k} = True$
      count($events_i^{j_k}$) = 0

would be detected) but its location estimation is coarse-grained. In contrast, the Delta algorithm easily pinpoints which transmitter-receiver link the person crosses, but it loses track of a person that does not cross a light barrier link. To obtain both large coverage and fine-grained localization, one can combine the results from both algorithms, for example, by calculating the centroid of their estimated locations.

4.5 Room Activity and Occupancy Recognition

In this section, we introduce the room activity recognition and occupancy classification module. The study focuses on a conference room, with activities and occupancy levels categorized as in Table 4.1. This module uses a supervised machine learning approach based on a feature vector of light power measurements. For other types of rooms, our activity classifier needs to be trained separately to classify different set of activities that commonly happen in these rooms.

The features to be used have to cover all the room's different activity spots, thanks to the non-LOS nature of shadow based tracking. Based on our hypothesis, detecting the room's occupancy and different activities can be inferred from the sources of movements and light

Table 4.1: Room activity and occupancy categories

Activity		**Occupancy**	
Index	Room Activity	Human Count	Category
0	Empty Room	0	Empty Room
1	Sitting at/near Table	1	Single Person
2	Whiteboard Discussion	2-3	Few People
3	Projector Presentation	> 3	Many People
4	Single Person Rehearsing		
5	Conducting Experiments		

settings at different locations. For example, during a presentation activity, the ambient light is usually dimmed and most light received is coming from the projector. One can think of using the delta values and base light level readings during OFF phase of the transmitter for the feature vectors. However, limiting the features to only these two values might cause losing information needed for the classification. Also, the effectiveness of these features depends directly on the base light level, that may change from time to time. Therefore, to capture the temporal and spatial variability of light settings, we use the readings from all the receivers in the room; values for each receiver are 12 average readings of 6 timeslots (including ON and OFF phases). We include the readings from all the time slots since this enables our system to distinguish the source of the lights from multiple directions. The readings are averaged over a span of time window w. We choose w to be long enough to capture the different activities and movements by users indoors. Since humans walk on average 1.4 m/s [110], we vary this time window from a second to a minute long. We only report the time window that maximizes the classification accuracy.

Our activity and count recognition approaches uses ensemble learning, specifically AdaBoost.M2 [111]. In Adaboost, the classification results of other learning algorithms ('weak learners') are combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is able to tweak adaptively the weak learners without prior knowledge about their performance. We use regularized linear discriminant analysis (LDA) learners as weak learners. We train the room activity and occupancy ensemble classifiers with the feature vectors labeled with the activity index and occupancy category label, respectively.

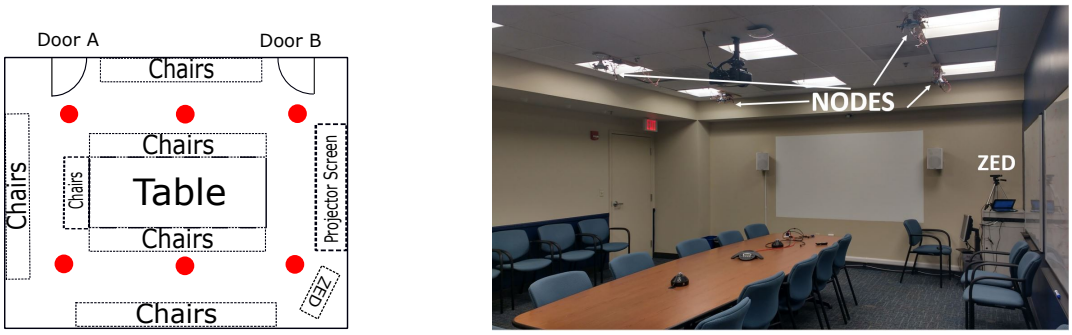
4.6 EyeLight prototype and testbed

In our prototype, we use an off-the-shelf Ecosmart 65W BR30 LED bulb as the transmitter for each light node. This light bulb contains an AC-to-DC module to provide DC power source to a series of LED chips. For our experiments, we remove this AC-to-DC module and feed 40V DC source directly from a DC power supply to the LED chips. We use a microcontroller (MSP432) to control a power MOSFET (IRFL520) as a switch to drive much larger current needed for the LED lamp. For *timeslot assignment*, to support 6 nodes, we divide each cycle (8ms) into 6 even timeslots.

In the transimpedance amplifier, we use the LF356 op-amp [112] which has low input noise voltage ($12\text{nV}/\sqrt{Hz}$) and suitable for photosensor amplifier task. The feedback resistor is $10\text{M}\Omega$ to maximize the transimpedance gain. In the later stage, further amplification is achieved by using INA126 [113], an instrumentation amplifier with low noise characteristics.

We use TI MSP432 Launchpad [114] to control both transmitter and receiver operations. The MSP432 Launchpad also offloads data measurement through Wi-Fi to our processing server with the help of a TI CC3100 BoosterPack [115].

We built 6 light nodes and placed them inside a conference room (size $7.5\times6m^2$, ceiling height $2.74m$), as shown in Fig. 4.5. All circuit components for each light node , including the microcontroller (MSP432 Launchpad), receiver boards, synchronization board, power board, were placed on a woodplank together with the LED light bulb. We placed 4 receivers around each LED bulb, pointing to different directions; each photodiode is titled $\theta = 10^0$ compared to the vertical line. This placement of photodiodes increases the number of virtual light barriers in the room to detect human presence. To construct groundtruth, we placed a ZED depth camera [116] in the corner of the room. The camera records videos of the room with depth information, and these videos are later manually processed to rebuild the positions of all persons inside the room.



(a) Room topology.

(b) Conference room.

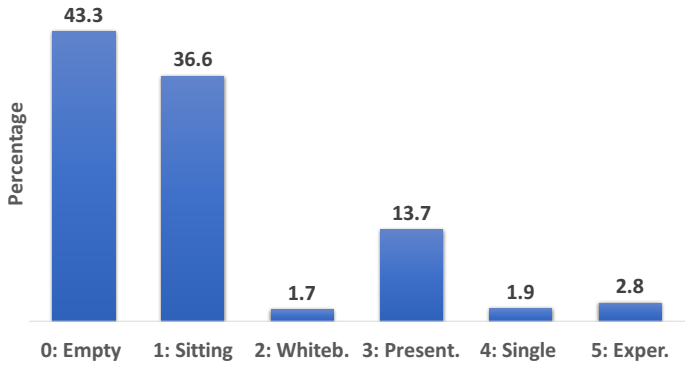
Figure 4.5: EyeLight testbed. There are 6 light nodes with distance between adjacent pair is 2.5m. The room has a central table, a number of chairs, and a projector screen.

4.7 EyeLight evaluation

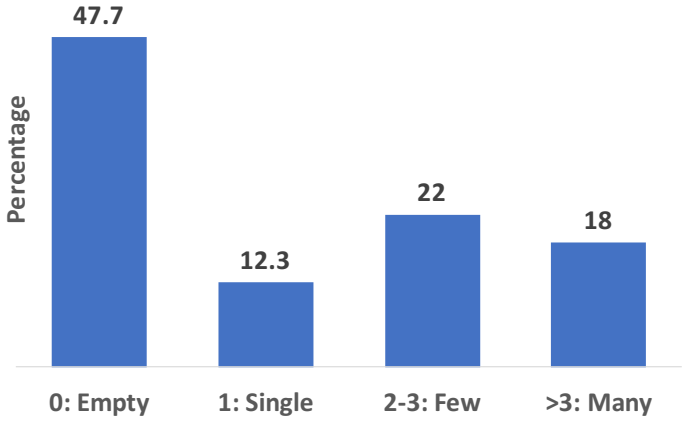
We collected data using our testbed in a conference room for 5 days over multiple weeks. For each day, we recorded data during normal working hours, the total number of hours recorded being 28.5 hours. Different users entered the room, including visitors, staff, faculty and students. Different lighting settings and different chairs organizations have been conducted during these days. We collected the base light level for the Spike and Delta algorithms at the beginning of each day.

4.7.1 Light barrier crossing detection accuracy

The output of the Delta detection algorithm for each photoreceiver is a binary detection: for each second, whether there is shadow casted by the adjacent node on the floor where the receiver is looking at. To evaluate the accuracy of our Delta detection algorithm, we conduct an experiment in which several test subjects walk in the room across all the lamps. Fig. 4.7 shows the True Positive Rate (TPR) and False Positive Rate (FPR) of the delta detection algorithm for different photoreceivers. TPR is the ratio of the correctly detected events over the total number of proximity events, and FPR is the ratio of the incorrectly detected events over the total number of testing cases when no person is in the vicnity of a sensor. The receivers in the figure are the ones pointing to an adjacent light bulb. The average TPR across all receivers is 82.17% and the average FPR is 5.77%. Among all receivers, only receiver 5.3 has low TPR (6%). Given our conference room has dark



(a) Activity Identification



(b) Occupancy Estimation.

Figure 4.6: The distribution of different activities and occupancy categories in the dataset.

carpet with low reflected light, the TPR and FPR value reported here are reasonably good. Also, this is the performance for each single receiver; we expect that by combining multiple receivers together, the accuracy of the whole system would be higher.

4.7.2 Localization error

Fig. 4.8 shows the localization error for single-person tracking scenarios, using three different methods: using only spikes detection, using only delta detection, and combined. Delta detection shows lower localization error (median 0.89m and 90 percentile of 2.5m) than spikes detection (median 1.18m and 90 percentile of 2.56m). However, the spikes detection is achieving this localization error while covering 94% of the time in which the user is inside the room compared to 69% for the delta detection. It is clear that there is a tradeoff here

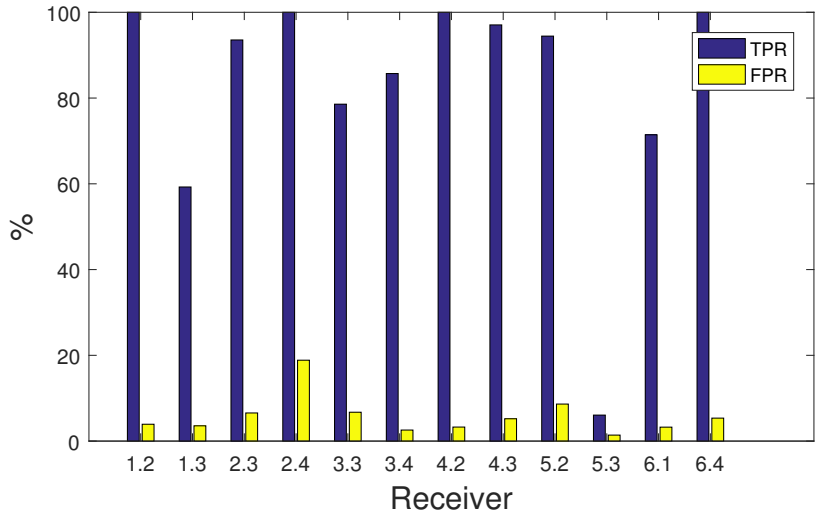


Figure 4.7: True positive rate and false positive rate of delta detection algorithm for all different sensors.

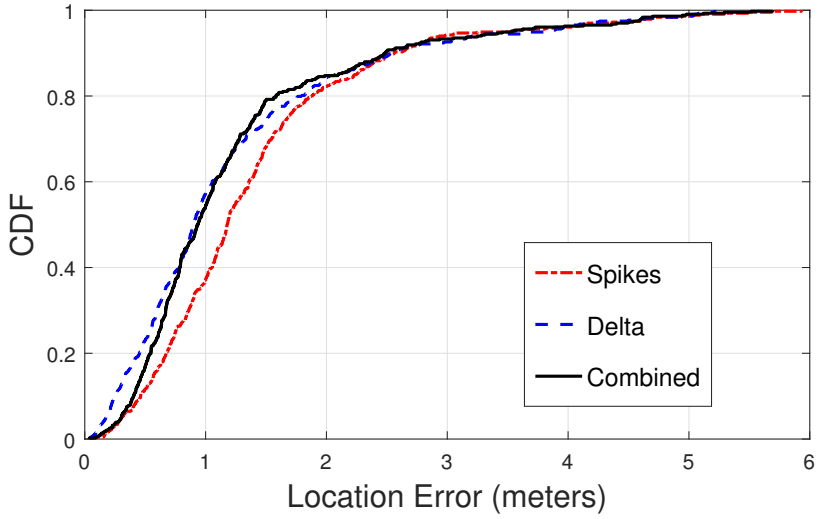


Figure 4.8: CDF of localization error for three cases: 1/ using only spikes detection, 2/ using only delta detection, and 3/ combined detection.

between the coverage and localization accuracy. Therefore, we also propose the combined version of the two algorithms, which achieves a 0.94m median error and better coverage rate than the delta detection.

4.7.3 Room Activity Recognition and Occupancy Estimation

We evaluate our room activity recognition classifier by 10-fold cross validation over the whole collected dataset using random partitioning. Each feature vector represents the average

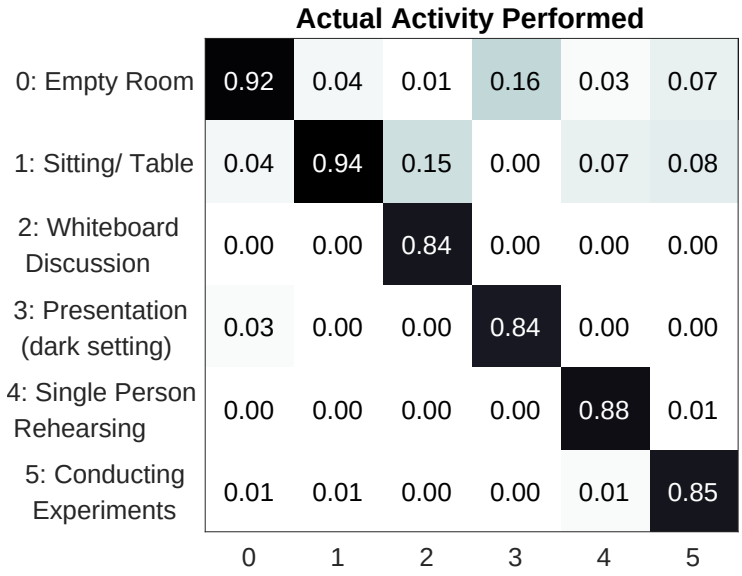


Figure 4.9: Confusion matrix for activity identification in conference room. Total size of the dataset is 102889 feature vectors, corresponding to 28.5 hours.

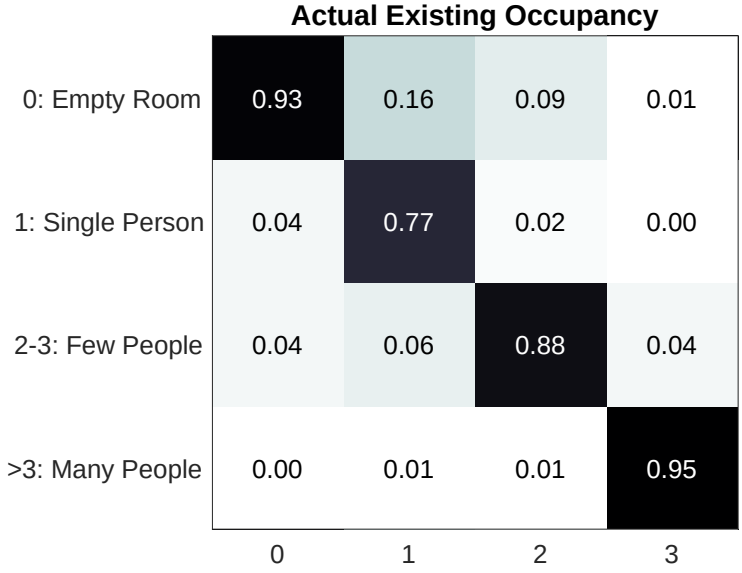


Figure 4.10: Confusion matrix for occupancy estimation in conference room. Total size of the dataset is 1710 feature vectors, corresponding to 28.5 hours.

readings over a 5-second period, which maximizes the classification accuracy. Fig. 4.9 shows the confusion matrix for the classification results of our activity recognition classifier. Each column represents the actual activity performed by the user and each row shows the activity as classified by our system. The overall classification accuracy is 93.78%; however,

if we break down the TPR for each activity, we can see the performance degrades for categories 2: whiteboard discussion, 4: single rehearsal and 5: conducting experiments. These activities represent a small fraction of the collected data as presented in Fig. 4.6a, and therefore, the classifier likely has not enough data to accurately capture the true model of these classes. Also, class 3, presentation in the dark, is easily misidentified as class 0, empty room, since the room is almost dark, and during presentation there are not many movements to capture. However, we expect collecting more data specially for these classes will decrease the classification error.

EyeLight is able to distinguish 4 classes of occupancy of a room, by classifying the readings coming from all the nodes inside. Each feature vector represents the average readings over a 10-second period, which maximizes the classification accuracy. We used the same evaluation procedure of the activity recognition classifier (10-fold cross validation). Fig. 4.10 shows the confusion matrix for occupancy estimation classifier. The overall accuracy of the classifier is 93.7%, while the TPR for single person class is lowest among all the classes with 86%. A single person staying in a conference room is not a common event, so the dataset for this class is not enough. Detecting a single person is thus more challenging than multiple persons specifically, since the collected data for single-person class is also the lowest among the four classes as in Fig. 4.6b. Moreover, a single person induces low effect on the light especially when not moving (e.g., sitting near the table and working with on a laptop). Therefore, we moved from five-second feature vectors, as in the activity recognition, to a ten-second feature vector in order to capture more of these rare movements for a single user. Again, we expect that collecting more data for this class would improve the classification accuracy.

4.7.4 Microbenchmark experiments

Distance between nodes. We increase the distance between the two nodes and measure the delta values received in one receiver for each distance between two nodes (Fig. 4.11a). As the distance between two nodes increases, the delta value becomes smaller; starting from 3.5 meters, this delta is too small to distinguish from noise, rendering EyeLight ineffective to use.

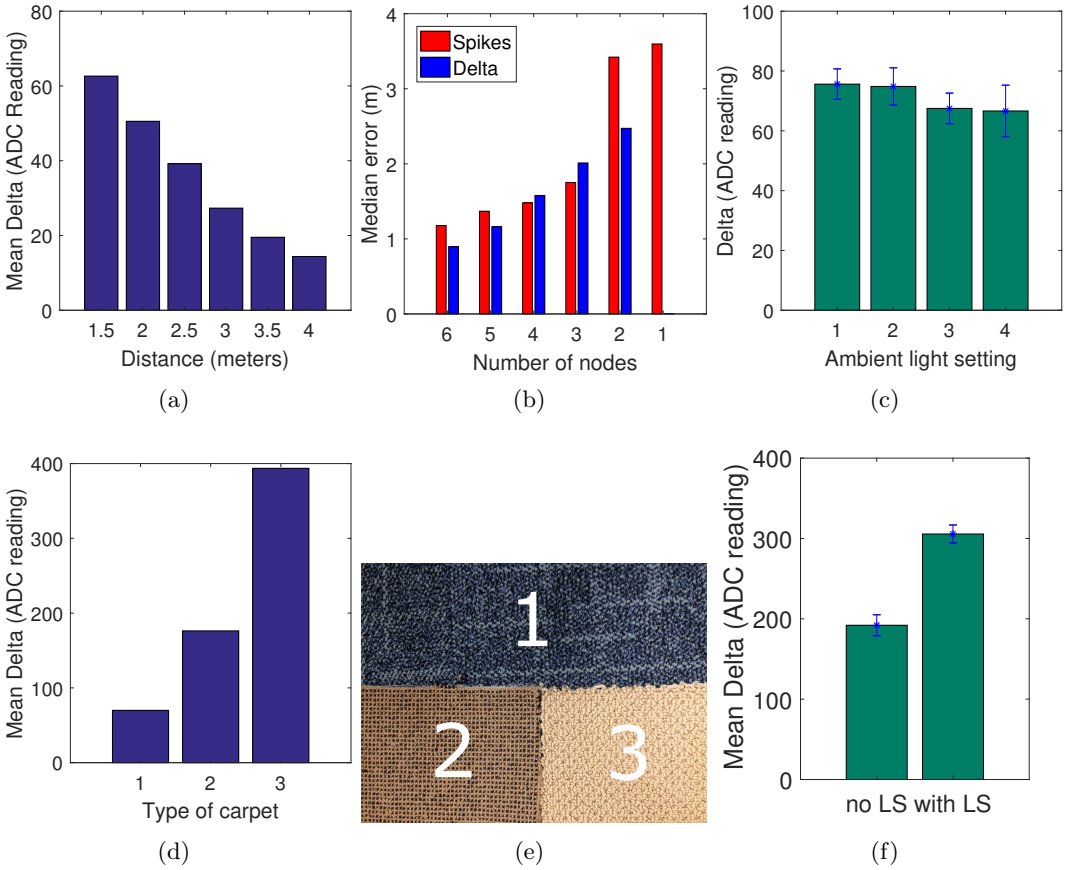


Figure 4.11: (a) Delta values for different distance between two nodes. (b) Location median error for different number of nodes. (c) Delta values for different ambient light settings. (d) Delta values for different types of floor carpets.(e) Types of carpet in (d). (f) Effect of lamp shade.

Number of nodes. We measure the localization median error of two algorithms (Spikes and Delta) with reducing number of nodes to cover our conference room (Fig. 4.11b). Note that there is no data for single node case of the Delta algorithm detection, since it needs at least a communication link between two nodes. As expected, the location accuracy reduces as the number of nodes decreases, because either the number of guarded locations (for Spike detection) or the number of virtual light barriers (for Delta detection) decreases.

Ambient light. We test different ambient light settings in our conference room: no ambient light, only ceiling lights turned on, only side lights turned on, both ceiling lights and side lights turned on. The mean and standard deviation of delta values for each light setting over a period of time is shown in Fig. 4.11c. For each ambient light setting, the standard deviation is small, allows the delta algorithm to work efficiently. However, the mean value

of deltas slightly differs between light settings, suggesting that the system might need to calibrate for several times a day, when the ambient light setting is changed.

Different types of carpets. Another factor that affects the efficiency of the delta-based virtual light barrier crossing detector is the reflectivity of the floor carpet materials. The carpet inside our conference room is dark, and thus reflects less light. To see the applicability of our detection algorithm on other types of carpets, we tested a light node facing different types of carpets (Fig. 4.11e) and compute the delta values (Fig. 4.11d). As can be seen, two other carpets have brighter surface, giving much larger delta values. Therefore, we believe EyeLight is also suitable to work with other room carpet, with even better performance. For other floor types, such as tiles, wood, due smoother surface, they reflect light even better than carpets, thus are also applicable in EyeLight.

Lamp shade. In all previous experiments, we tested with commercial light bulbs without lamp shades. To show the effect of lamp shade on the transmitted signal, we compared the average delta values for lamps with and without lampshade (Fig. 4.11f). The result shows that with lamp shade on, the average delta value actually increases. One might think that lampshade would reduce the intensity of the light from the transmitter, weakening received light power at the receiver. In fact, however, the lamp shade distributes light more evenly over the floor area under the lamp, thus improve the sensitivity of the receivers looking at different spots on the floor.

4.8 Discussion and Conclusion

We proposed a device-free indoor tracking, occupancy estimation and activity recognition system that can be integrated in light-bulbs. The key idea is to create a mesh of reflective virtual light barriers across networked light bulbs to detect occupant movement. We found that our high-sensitivity photo-sensing circuit can detect minute light changes (shadows) even on dark carpeting, and that a time division pulse signaling scheme allows differentiating the light nodes causing shadows on the floor. With our 45 m^2 conference room prototype system with 6 light bulbs each carrying 4 receivers, we further found that the sensing system can achieve a 0.89m median localization error as well as 93.7% and 93.78% occupancy and

activity classification accuracy, respectively.

Our current system still has several limitations that could be addressed in future work. *First*, EyeLight requires more than one lamp per room for fine-grained user tracking. Fortunately, the small size of LED lights makes it easier to add additional lights in rooms. *Second*, EyeLight so far focuses on tracking a single person per room. It could track multiple persons as long as they cross different virtual light barriers, while multiple persons walking together leads to mixed shadows. *Third*, EyeLight needs to adapt to different light settings, such as different times of the day, rooms with outdoor light passing through windows. Currently our prototype works in a conference room without windows, where measured illuminance of light reflected from the floor is under 5 lux. In a room with outdoor light entering through windows, the current receivers saturate. However, techniques like Adaptive Gain Control, as used in other systems dealing with high dynamic range, can be added to EyeLight to improve its robustness. An adaptive system is also needed to keep track of the change of the baseline light level. We leave such designs for future work.

Chapter 5

Capacitive Coupling-based Micro, Dynamic Finger Gesture Recognition

5.1 Introduction

Head-mounted devices (HMDs) for Augmented Reality (AR) are transforming modern workspaces thanks to their ability to overlay digital information onto the physical world. There are a growing number of applications of these devices in different industries, such as image-guided therapy [117], site productivity improvement for construction workers [118], online support for field service workers [119], training new employees [120]. However, providing inputs to these devices while being user friendly, intuitive and ensuring an immersive experience remains a challenging problem: the current input techniques mostly require users to hold a tablet or smart phone in one hand or both hands or require hands to be present in the field of view of a sensor. This often leads to inconvenient interactions and limits the device usage in mobile scenarios. For example, camera-based detection of in-air gesture interfaces (Microsoft HoloLens [121]) requires users to raise a hand to eye level, which can cause fatigue over longer periods of use and is also impractical in some scenarios, such as repair and maintenance. Voice input can be convenient for some simple instructions or information, but can be disturbing in a common workplace setting. Therefore, to advance the usage of the head-mounted devices, an *always-available*, *low-effort*, and *expressive* input method is required.

Input methods using hand or finger gestures can satisfy this need. Current techniques being used for hand/finger gesture recognition include off-body sensing (cameras [121], radar [122], acoustic [123]) and on-body sensing (inertial sensors [124], impedance tomography [125], magnetic sensors [126]). Some approaches seek to reconstruct arbitrary hand

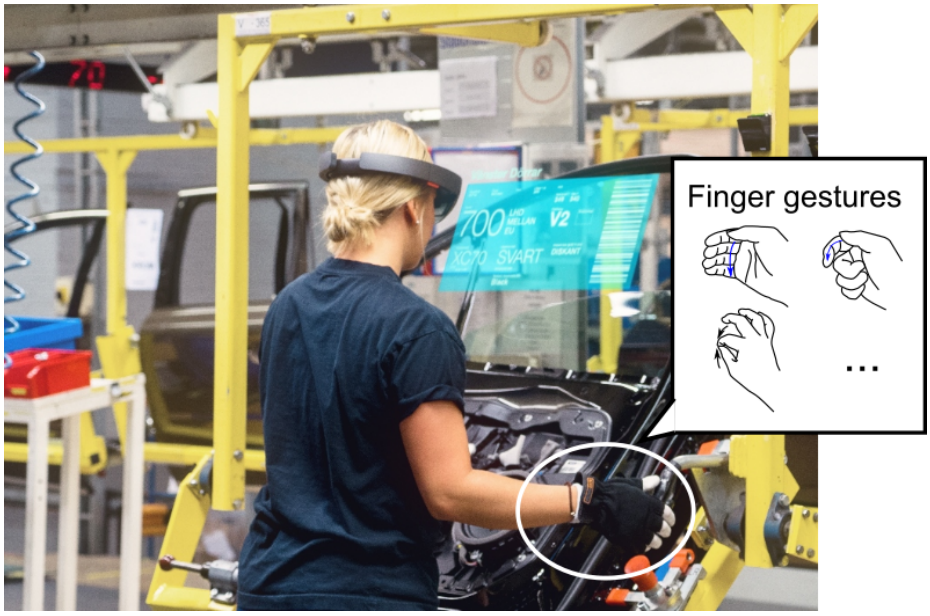


Figure 5.1: **HandSense concept**. While Augmented Reality head-mounted devices start to find applications in areas like manufacturing, repair and maintenance, providing inputs for these devices with low-effort from users remains a challenge. HandSense offers an always-available, user-friendly dynamic, micro finger gesture recognition system for these devices.

poses, but generally rely on cameras, which require the hand in the field of view and significant computational overhead, or visible light sensing [127], which also requires user hands to be inside the sensing space. In addition, gestures being recognized often include large movements of the fingers or the whole hand, which can be tiring to users during/after long periods of usage. The existing gesture recognition techniques fall short of satisfying the needs for controlling HMDs in working environments because of the following reasons: unable to operate outside a specific region of sensor operation; heavy instrumentation on the hand or in off-body sensors; difficult to detect low-effort finger gestures that are more suitable for HMD controller.

In this work, we propose HandSense, a *light-weight*, *always-available* system to recognize a series of *dynamic*, *micro* finger gestures that are highly suitable for controlling HMDs. The key idea in HandSense is measuring and classifying pairwise profiles of capacitive coupling between electrodes placed on all fingertips. Capacitive coupling between two electrodes is a monotonic function of the distance between them; it therefore allows inferring distance between two corresponding fingertips. Given the structural constraints of the human hand,

the inferred fingertip distances allow recognizing micro finger gestures.

This approach is motivated by the observation that there exists a large and important class of augmented reality applications where users typically wear gloves (e.g., in the medical, maintenance / repair, manufacturing, or certain e-sport domains). The electrodes can be integrated into the fingertip sections of such gloves, akin to how many gloves already contain conductive materials at the fingertip to enable touchscreen use. Placing electrodes on each fingertip can therefore be much less intrusive than one might initially assume. Note also, that in contrast to more heavily sensor-instrumented gloves for sensing hand motions, such as DataGlove [128] or fiber-optic gloves for VR applications [129], HandSense only requires electrodes as sensing elements, which can be fashioned from cheap conductive materials such as copper tapes or conductive thread (connecting to an external processing unit possibly placed inside user's smartwatch or a wristband), thus the gloves can be particularly useful in medical or high wear and tear working environments. While currently intended for gloves, advances in skin electronics [130] (perhaps electrodes and traces back to the on-wrist device) may allow HandSense techniques to be used even in applications where users do not wear gloves. Overall, note that electrodes and traces are not active components, thus the fabrication can be low-cost.

Another aspect that helps HandSense better serve as a gesture controller for HMDs is its low-effort, always-available property. Since the system relies only on interactions between sensing elements on fingers, it is not limited by the working range or suffered from occlusion from external sensors (e.g. cameras [121], radar chip [122]). In addition, HMD users in working environments often have their hands occupied (e.g. therapists working on medical devices, cargo workers holding packages); in these cases performing finger gestures with small movements in any place is the more preferred method over whole-hand movement onto the virtual dashboard, which is inconvenient and interrupting to the workflow.

There are several challenges in realizing the HandSense system. *First*, the human hand is a large conductive surface, thus the dominant capacitive coupling of the fingertip electrodes is through the hand and the signal is much less dependent on the relative distances between the electrodes. To further increase the dynamic range of the detection of spatial relationship between electrodes, we seek to reduce the unwanted influence of the hand through the use of

an additional *ground electrode* on each finger. *Second*, to be able to detect quick, dynamic, micro finger gestures, the capacitive coupling measurements should be fast to provide frames of link measurements quickly. We use *synchronous undersampling* technique, which is a light-weight, low complexity method for estimating the received signal amplitude. *Third*, as over-the-air capacitive coupling between finger electrodes decreases quickly with distances, the link measurements between non-adjacent fingers are less usable in the capacitive profile. We identified an additional *through the hand* capacitive coupling path between all fingers, thus enabling more types of finger gestures to be recognized.

In summary, the major contributions of this work are as follows:

- Proposing a placement configuration for electrodes on fingertips to enable measurement of capacitive coupling between each pair of fingers with minimal effect from user's hand.
- Designing a light-weight capacitive profiling system for measuring pair-wise capacitive coupling between fingers, which are then used for finger gesture clasification system.
- Identifying three types of finger interactions detected by the capacitive profiling system, which enable more dynamic, micro finger gestures to be recognized.
- Designing and implementing a glove prototype and evaluating HandSense in recognizing a set of 14 different micro finger gestures based on data collected from 10 subjects.

5.2 Background

Current modalities of interacting with computers, mobile phones, laptops, tablets and smart watches is by using keyboards, mice, trackpads and touch screens. But with the advent of Augmented Reality (AR) and Virtual Reality (VR) platforms these existing modalities of human computer interaction fall short from the aspects of immersion, ease of use and being intuitive. A big part of crafting such an experience lies in how easily and seamlessly the user is able to interact with the virtual environment. To this end it can be argued that enabling a user to interact with the AR / VR environment directly with their hands would

ensure a more intuitive and immersive user experience. In order to accomplish this, the AR / VR systems need to be able to recognize what exact gestures users are making to interact with them. From a VR perspective the user is not able to see their hands but the exact hand position and finger configuration need to be rendered with high precision. Whereas, from an AR perspective the user will be able to see their hands but would need to interact with overlaid virtual interfaces.

5.2.1 Existing finger gesture recognition techniques

Data Gloves: Gloves have been used to detect hand gestures since the early seventies. They are a reliable way of sensing the gestures/hand movements that the user is making. Early gloves such as the Sayre Glove [129], Data Glove [128], MIT LED glove [129] and CyberGlove [131] were sensor dense and usually had a mix of flex/bend sensors, touch sensors, inertial motion sensors, tilt sensors, ultrasonic sensors, LED and photosensors sensors. These sensors were mostly affixed to the glove. This class of "data gloves" were rich in providing data generated from different parts of the hand. However, the sensors were not cheap and due to their large numbers the gloves were bulky, cumbersome to carry and usually restricted the movement of the users hands. The sensors could not translate small changes in flexion to finer or dynamic gestures. They also usually required a user-specific calibration procedure.

Camera-based approach: Another approach is to capture the movement of the users hands directly using a camera and then inferring the gestures made. This is demonstrated by HoloLens [121], DepthTouch [132], 6D Hands [133], Keskin et al. [134] and Microsoft Kinect [135] where captured raw images/video of the hand are processed using sophisticated computer vision algorithms to determine the hand position and gesture being made. On the front end this method eliminates all burdens from the user to wear or carry extra devices. It also makes the experience of using the system immersive and intuitive as users are now able to interact with the system with their bare hands. However, this approach assumes that the hands can continuously be monitored, are always in the field of view of the camera (i.e., no occlusion) and external illumination conditions will always permit capturing data of satisfactory quality. On the back-end this method requires the availability of a powerful

enough computer that can run these sophisticated algorithms to process the acquired raw hand tracking images. Finally, housing both the camera and the computer often leads to bulky devices / systems.

WiFi, radar and light-based approaches. Other modalitiesused to perform hand gesture recognition are WiFi channel state information (CSI) [136], WiFi received signal strength (RSSI) [137], shadows cast by visible light [127] and more recently radar based systems [122]. Deployed WiFi access points (APs) leverage changes in channel state information and drops in received signal strength to detect hand gestures. Visible light based techniques infer 3D gestures based on shadows that are cast on photoreceivers. Google's Project Soli [122] developed a 60GHz radar chip that is able to detect micro movements of fingers. The chip pings a signal similar to a radar and looks for reflections, hand gestures are determined by feeding these received pings to a trained random forest classifier.

HandSense approach. A common aspect of the systems discussed earlier is that all of them expect that hands be in the field of view (FoV) of their deployed sensors. They also require the surrounding environment of the hand to be fairly stable (e.g. availability of sufficient light, not too much movement around the subject). These systems are unable to sense micro gestures or dynamic movement of fingers. For these reasons these modalities fail to be good options for use in conjunction with head mounted displays in active work/industry environments.

HandSense is able to overcome these problems by making use of capacitive sensing. This sensing modality has the following advantages: 1) HandSense is *always available*; it infers the relative spatial relationship between fingers, hence, the hands can be anywhere and gestures can still be recognized, 2) by its sensitivity to close range movements, capacitive sensing allows recognizing micro finger gestures, 3) sensing electrodes are cheap and the glove can be light-weight when embedding electrodes in it, and 4) low computation overhead.

5.2.2 Capacitive sensing

Capacitive sensing is an ubiquitous sensing technology in human-computer interaction. It works by measuring the capacitance variation between two or more conductors. In the most basic form, the capacitance between two parallel plate conductors is $C = \frac{\epsilon_0 \epsilon_r A}{d}$, where

ϵ_0, ϵ_r are the free space and relative dielectric constants, respectively, A is the area of the conductor plate and d is the distance between the two conductors. While there are many forms of capacitors, the capacitive coupling between two conductors is always affected by only these three factors: electrode size, dielectric between electrodes and distance between them.

The measurement technique in HandSense is closely related to the shunt-mode capacitive sensing [138]. In this mode, a capacitive link is established between two electrodes, in which one electrode is powered by an AC signal and the displacement current is measured at the other electrode. The displacement current is proportional to the capacitive coupling amount between the two electrodes. Each sensing electrode can be configured as either a transmitter or a receiver. For n electrodes, there are $\frac{n(n-1)}{2}$ distinct measurements for all transmitter-receiver combinations. Note that the electrodes are in fixed positions with careful calibration to better detect the appearance/position of human body parts.

While also using excitation-response measurement approach as in the shunt-mode method, the electrodes in HandSense are placed at *mobile* positions; in particular on fingertips. It then uses the pair-wise capacitive coupling measurements between electrodes to infer micro gestures performed by users. Measurement with this particular electrode placement presents both challenges and opportunities: on one hand, it is difficult to calibrate the measurement system with mobile electrodes, and the large surface of the user's hand causes most capacitive coupling between electrodes to pass through the hand. On the other hand, distance between fingers when performing gestures is small enough for a capacitive coupling measurement to work. Also, the relative motion between fingers is constrained (fingers can only flex and move in certain directions). HandSense optimizes the electrode placement to only expose the capacitive coupling path associated with finger gestures, including close-range over-the-air coupling and *intended* through-the-hand finger communication, while reducing the *unwanted* coupling in the back channel between electrodes.

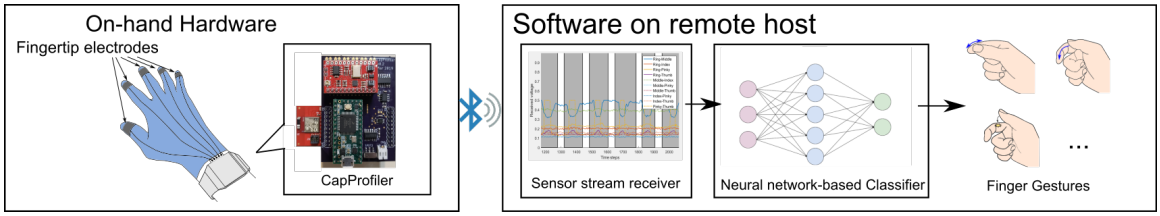


Figure 5.2: System overview

5.3 HandSense overview

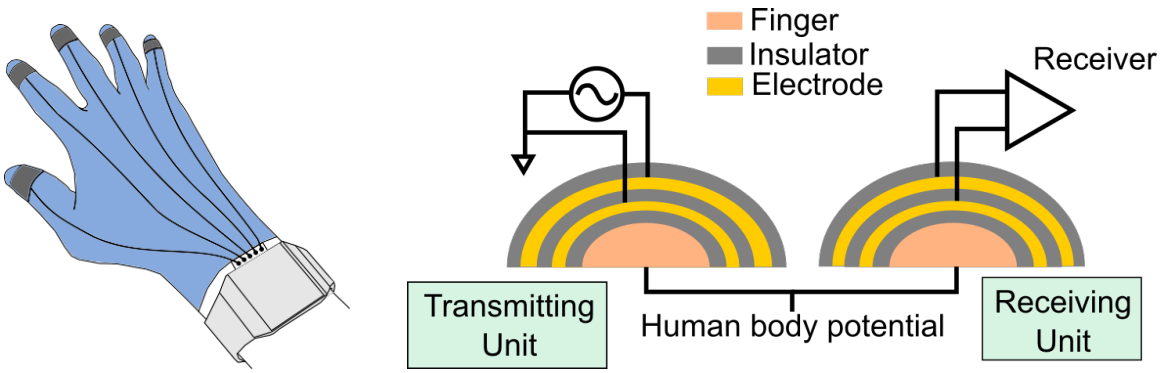
HandSense is able to recognize a set of dynamic, micro finger gestures that are suitable for use in conjunction with head-mounted devices. This is enabled by placing electrodes on each of the five fingertips of a hand and inferring the spatial relationship between them through capacitive measurements from all the pairs of electrodes. HandSense, therefore, is self-contained: unlike approaches using cameras, on-body or external RF sensors, HandSense is able to detect finger gestures when the hand moves anywhere in space, even when it is not in the field of view (FoV) of a head-mounted devices or hands are occluded. The system is also able to detect fast movements (comparable to the speed of a quick swipe), thus allowing the gestures to be low-effort to users. The availability everywhere and the ability to detect fast, low-effort gestures make HandSense a highly suitable input method for head-mounted devices.

On-finger electrode design. A simple method to infer close distance / proximity between two electrodes in free-space is by measuring the capacitance between them as capacitance is inversely proportional to the distance between electrodes. However, a naive configuration of affixing one electrode on each finger comes with problem: a large amount of the coupling between the electrodes would be through the hand as opposed to over the air. This is because the hand is more conductive than air and most of the capacitance coupling between the two electrodes would be through the lower impedance path along the hand. Hence it becomes difficult to measure the small change in capacitive coupling through the air on top of a large capacitive coupling through the hand when the fingers move closer or further away from each other. To solve this problem, we propose adding a ground electrode underneath each signal electrode to minimize the capacitive coupling between the signal electrode and the user's hand. More discussions about this design is in Section 5.4.

Minimally instrumented glove design. HandSense consists of a central controller board worn on a user's wrist and a glove which is used to equip the user's fingertips with sensing electrodes. Note that the glove only requires passive components; electrodes and traces. This makes HandSense particularly useful in high wear and tear environments, such as healthcare, wellness and fitness, automobile/factory shop floor, assembly line. A user can connect their own smartwatch/wristband with a new glove to use with his head-mounted device.

Light-weight pair-wise capacitive coupling measurement techniques. HandSense is based on the insight that most finger gestures can be inferred from a profile of pair-wise capacitive coupling measurements between fingertip electrodes. Furthermore, since HandSense seeks to recognize dynamic finger gestures, the pair-wise capacitive coupling profile contains not only measurements at one instant in time but a time series of measurements, providing richer data for finger gesture classification. For typical dynamic, micro finger gestures (e.g. sliding, tapping), which can last under 1 second, the measurement system should repeatedly sample all finger-pairs fast enough to deliver sufficient data points to infer the gestures. We employ several techniques to satisfy this requirement: (a) fast switching between electrodes to act as transmitters and receivers, (b) a *synchronous undersampling* measurement technique to quickly estimate the instantaneous received signal in each link. The synchronous undersampling technique is light-weight in both hardware and firmware: it avoids the needs of expensive components such as mixers, phase shifters, and low pass filters, as in the traditional synchronous detection technique. The on-board firmware only requires a moderate ADC sampling rate and minimal computational overhead, as opposed to the Discrete Fourier Transform technique. Such low requirements made it easier to integrate the controller electronics into low-cost wristbands for HMD users. We describe these techniques in more detail in Section 5.5.

Finger gesture recognition. With the above electrode placements and measurement techniques, we describe three different finger interactions that HandSense can recognize: direct over-the-air finger proximity, finger touching, and indirect through-the-hand electrode communication. These finger interactions produce signal signature in the time series data, which can be used for recognizing more finger gestures. With this time series data of



(a) A glove with electrodes attached on fingertips and connected with a wrist-worn device.

(b) Cross-section view.

Figure 5.3: Electrode placement.

measurements on pair-wise links, we investigate different neural network based techniques to classify the finger gestures. More details are in Section 5.6.

Design overview. Fig. 5.2 illustrates the overall design of HandSense. The sensing electrodes are attached on a glove at the fingertip sections. These electrodes are connected with a central controller board, called *CapProfiler*, which could be embedded inside a smartwatch or wristband. Inside this board, a microcontroller controls the signal transmission through a signal generator module, receives signal from an analog receiver circuit, and coordinates timing of different transmitter-receiver links. Received signal amplitudes calculated from the measured signal on all links are packaged into frames and sent over Bluetooth to a remote host, which can be a head-mounted device the user is wearing. The time-series signal sequences of all the communication links are then passed through a trained end-to-end neural network model to classify into different finger gestures.

5.4 Design of on-glove electrodes

The electrodes (conductor plates) act as both transmitting and receiving elements in HandSense. They are placed on the top bone (distal phalanges) of each finger (Fig. 5.3(a)). The rationale for placing the sensing electrodes in these positions is that the fingertips are the most active parts of the hand, and they take part in almost all gestures. While we seek to measure the capacitive coupling between each pair of electrodes over the air (small dielectric), the higher dielectric constant capacitive coupling path through the hand presents a

challenge to HandSense. In addition to acting like a resistor, the outermost layer of skin (epidermis) acts like a capacitor if placed in contact with a piece of metal. The underlying tissue represents one plate of a capacitor and the metal surface the other. The dry epidermis represents the less conductive material or "dielectric" in between. In our case we use an AC source to excite the electrodes, this AC source "shorts" out the natural resistance of the epidermis allowing the current to bypass that part of the hand's resistance and making the hand's total resistance much lower. This resistance further reduces with increasing frequency of the current. This means that the dominant signal path goes through the hand (the less resistive path) as opposed to through the air. According to the National Institute for Occupational Safety and Health (NIOSH) the resistance offered by the human body is in the range of 1000 to 100,000 Ω [139] and the capacitor with $A = 2cm^2$, $d = 3cm$ with air as dielectric has a capacitance of 590pF and an impedance of 26MΩ [140]. Hence a weaker amount of capacitive coupling over the air between two signal electrodes in the presence of a stronger capacitive coupling through the body would be more difficult to measure. This is detrimental to our system as we wish to estimate the over-the-air distance between fingertips based on the capacitance between the fingertip electrodes.

To address this challenge, we place a ground electrode between each signal electrode and the finger, with insulation layers in between to prevent shorting of the electrodes (Fig. 5.3(b)). It is evident from Fig. 5.4 that there is not much change in amplitude at 100kHz when the ground plane is absent, whereas in the case with the ground plane we can see that there is a drop in amplitude when fingers are moved apart to a distance of 3cm.

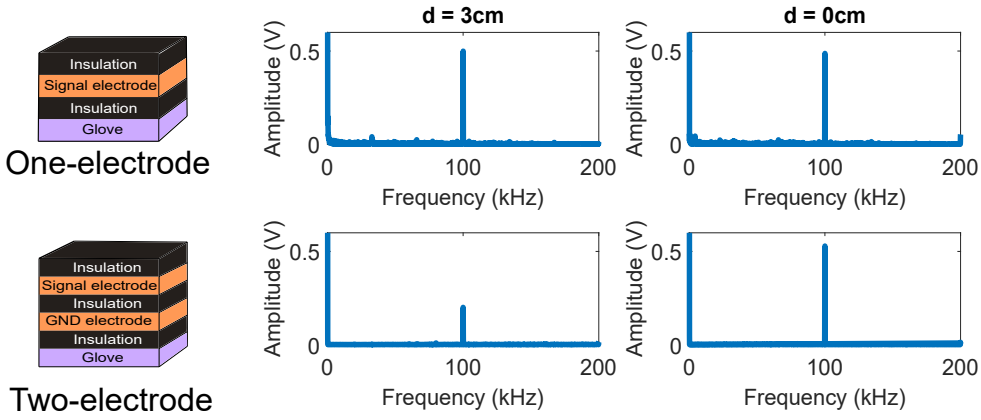


Figure 5.4: Received signal at 100kHz in one-electrode vs. two-electrode designs. Here d is the distance between the two fingers during its transmitting-receiving session.

Adding a ground electrode underneath the signal electrode closer to the signal electrode than the skin helps, as it is at a lower potential than the skin. Hence it couples stronger with the signal electrode. It also provides a common ground to the smart watch/device which measures the voltage. Without the ground electrode, the transmitting signal would couple to the user's hand and then couple to the receiving electrode.

Note that there are other advantages in having a defined ground electrode: 1) the ground plane ensures that the signal is always coupled to the same ground potential across all fingertips as each fingertip has the same ground electrode underneath the signal electrode. Without this common ground, each signal electrode is coupled to its own dynamically changing finger potential, 2) generally frequency multiplexing (i.e., each finger is assigned a pre-determined frequency of operation) techniques are used to uniquely distinguish received signals from different fingers. But since we are measuring distance using capacitance, fingers that are far away from each other produce signals that have very low amplitude. Having a common ground plane ensures that the calculated signals are also with respect to the same potential which means that all the fingers can be excited with the same frequency signal.

5.5 Design of the capacitive profiling system

The central controller board of HandSense, called CapProfiler, can be embedded inside a wrist-worn device such as a smartwatch or wristband, which leaves only electrodes on the

glove. CapProfiler board follows modularized design: signal excitation, reception, as well as signal processing are all integrated on board, and the system can be put to use once the user connects glove with sensing elements with the CapProfiler board. To further lower the cost of making CapProfiler boards, we seek a design with low complexity hardware and light-weight measurement techniques in firmware.

5.5.1 Transmitter and receiver design

Capacitive Coupling Transmitted Signal. At any given time, HandSense transmits a sinusoidal wave as an excitation signal to an electrode and measures the received displaced current from a nearby unexcited electrode to infer the capacitive coupling between the two electrodes. The choice of transmitting frequency is dictated by several factors. On one hand, as the impedance through the air between the two electrodes is $X_C = \frac{1}{2\pi fC}$, the higher the frequency is, the lower the inter-electrode impedance is, causing more displacement current at the receiving electrode. On the other hand, higher transmitting frequency requires higher ADC sampling rates and real time processing capabilities. In HandSense, we choose 100kHz sinusoidal wave as our excitation signal.

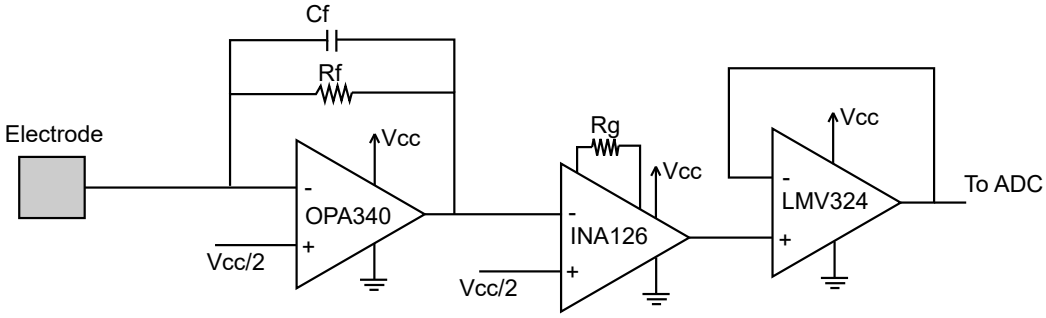


Figure 5.5: Analog receiver frontend

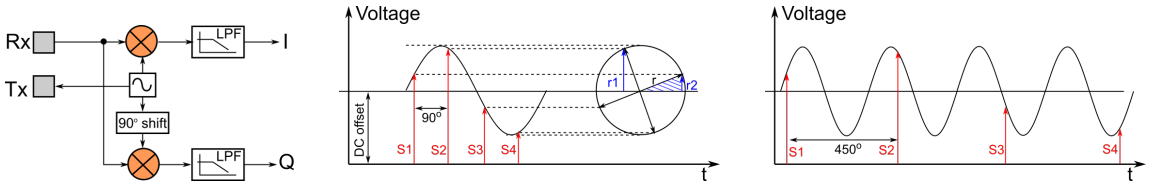
Analog receiver frontend design. We design a simple sensitive analog receiver frontend circuit connected to an electrode as shown in Fig. 5.5. The displacement current measured at the receiving electrode is amplified through a transimpedance amplifier. The amplifying gain of the transimpedance amplifier (OPA340) is set by the feedback resistor R_f following the formula: $V_{out}/I_{in} = -R_f$. It also has a capacitor in parallel to create a lowpass filter to filer out unwanted higher frequency components and harmonics. Since

the board is powered using a single supply a bias voltage of $V_{cc}/2$ is provided at the non-inverting terminal. This forces the DC output to about $V_{cc}/2$. The difference between this filtered, amplified output voltage and a bias voltage $V_{cc}/2$ is further amplified by a second stage using an instrumentation amplifier (INA126). This ensures that we amplify just the small received signal. The instrumentation amplifier has a default gain of 5 and additional gain can be set by using R_g. The output from the instrumentation amplifier is fed to a voltage follower with a low output impedance before being fed to the microcontroller ADC.

Multiple transmitters and receivers. We utilize a round-robin approach for multiplexing between different capacitive links, where one link is the capacitance between a pair of fingers (i.e., thumb to index finger is one link, thumb to middle finger is another link). We also observe as expected that links are symmetric (e.g., the middle-to-ring finger signal is the same as the ring-to-middle finger signal). The two multiplexers, one for transmitting and one for receiving, iteratively choose each of the electrode links, wait for the ADC to sample enough data points before switching to another link. By using the multiplexers we reuse the same signal generator and frontend receiver circuits, further simplifying our hardware design.

5.5.2 Estimation of received signal amplitude

A common technique to calculate the signal amplitude at a given frequency from ADC samples at a fixed sampling frequency is using Discrete Fourier Transform (DFT). However, this technique requires a high sampling rate, at least twice the frequency of interest, thus causing high processing overhead for the microcontroller. Moreover, we only need to compute signal amplitude for the transmitted frequency, thus most of the frequency spectrum produced by DFT is redundant. To avoid sampling data at high speed for high frequency signal (100KHz), we estimate the received signal amplitude by *synchronous undersampling* measurement technique, which was first proposed by Smith [138] and described in more detail in [141]. This technique can be seen as digital equivalent for synchronous detection method in analog domain.



(a) Synchronous detection. (b) Synchronous sampling. (c) Synchronous undersampling.

Figure 5.6: Measurement methods for estimation of received signal amplitude.

Algorithm 3 Calculation of received signal amplitude using synchronous undersampling technique.

Input: ADC sample array S (number of samples = 4n).
Output: Received signal amplitude.
$I = Q = 0$
for $i = 0 \rightarrow n - 1$ **do**
    $I = I + (S[4i] - S[4i + 2])$
    $Q = Q + (S[4i + 1] - S[4i + 3])$
$I = I/n$
$Q = Q/n$
return $amp = \sqrt{I^2 + Q^2}/2$

Synchronous detection is a common measurement technique for recovering the amplitude of the received signal at the transmitted frequency. Fig. 5.6(a) shows a typical hardware setup for the synchronous detection measurement method. The sinusoidal signal of frequency f from the transmitting electrode induces at the nearby receiving electrode a received signal consisting of attenuated version of the transmit signal plus noise. The received signal is multiplied with the original transmitted signal to produce sidebands at +2f and -2f frequencies and also a DC value. A subsequent low pass filter removes these sidebands, and the remaining DC value is proportional to the amount of displacement current on the receiving electrode. This assumes the phase of received signal and transmitted signal have the same phase. In practice, the received signal is demodulated with both the transmitted signal and its 90-degree-shift version, to recover the in-phase (I) and quadrature (Q) components. The received signal magnitude is then calculated as $\sqrt{I^2 + Q^2}$.

Implementing synchronous detection would require significant hardware cost (including mixers, phase shifters, and low pass filters). Moreover, the heavy low pass filtering after the mixer makes it slow to response to fast signal. *Synchronous sampling* seeks to remove these hardware components while still being able to estimate the amplitude of the received

signal at the transmitted frequency.

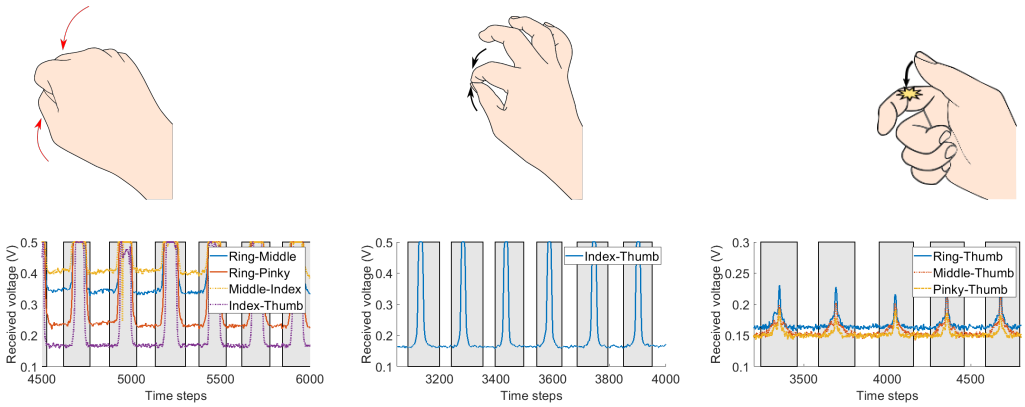
Fig. 5.6(b) shows a full period of a sine wave of frequency f with DC offset. If we sample at $4f$ sampling frequency, the 4 samples on each wave cycle are separated by 90 degrees each. Let S_1, S_2, S_3, S_4 be four samples on a wave cycle, when mapping these values onto an equivalent circle, we observe that $r_1 = |S_2 - S_4|/2$ and $r_2 = |S_1 - S_3|/2$. Applying Pythagoras's law for the shaded triangle, we also have $r = \sqrt{r_1^2 + r_2^2}$. This leads to the amplitude of the sinusoidal wave can be estimated as: $r = \sqrt{(S_1 - S_3)^2 + (S_2 - S_4)^2}/2$.

The synchronous sampling technique is fast: it only requires four samples to calculate the signal amplitude. However, it requires the sampling rate of $4f$, which can exceed the capability of some microcontrollers when the transmitted frequency is high (e.g. 100KHz). To reduce the required sampling frequency, we instead use *synchronous undersampling*. We assume that inside a small time window, signal is repetitive, so instead of sampling S_1, S_2, S_3, S_4 on the same cycle, we sample them on *continuous* cycles. Now the samples are taken 450 degrees each, and the sampling frequency can be reduced to $4f/5$. The formula to estimate the signal amplitude remains the same. To increase SNR, we accumulate values of S_1, S_2, S_3, S_4 over many cycles, average them before calculating the signal amplitude. Algorithm 3 shows the full procedure.

In our implementation, we use an ADC sample array of size 16 to calculate the received signal amplitude for each link. With sampling frequency of 80KHz, it takes 200us for capturing these 16 samples into a buffer. We implement ADC with Direct Memory Access, which frees the CPU from sampling process. In the main CPU process, we delay 1ms after switching the multiplexers to ensure the DMA buffer contains only samples after the link is stable. Therefore, a frame containing 10 measurements from 10 links takes 10ms, which leads to the measurement frequency of 100Hz in our implementation.

5.6 Micro dynamic finger gesture recognition

Typical finger gestures can be categorized into two groups: static gestures (such as making the victory sign, spiderman sign, okay sign) and dynamic gestures (such as swiping, sliding, tapping). HandSense focuses on the later group of gestures, especially the dynamic, micro



(a) Closed fist gesture. (b) Pinch gesture. (c) Tap gesture.

Figure 5.7: Illustrations of finger interactions recognized by HandSense.

gestures. These gestures are more suitable for interacting with the head-mounted devices for workers on manufacturing or construction sites: when the users hands can be busy with interacting with objects on the site, moving a few fingers to perform a gesture is less likely to disrupt the workflow. The gestures can be performed by finger muscles, as opposed to large hand muscle groups, thus reducing fatigue over longer use cases as well.

HandSense is highly suitable for detecting these type of dynamic, micro finger gestures. The system is capable of providing frames of link-wise measurements at rate of 100Hz, thus capturing more data points to recognize these fast, micro finger gestures. In addition, we realized that the finger movements in these gestures are more correlated with the *relative* position and velocity of each finger with regard to the other ones, as opposed to *absolute* position and velocity of individual fingers with regard to another coordinate system. Approaches using inertial sensors [124, 142, 143] or bend sensors [128, 131, 144] are able to track individual finger joints, but find it difficult to infer dynamic gestures being performed. Link-wise capacitive coupling measurements in HandSense provides better representation of these dynamic finger gestures.

In this section, we first show the three finger interactions that HandSense is able to recognize, then describe the neural network-based approaches to the problem of classifying finger gestures.

5.6.1 Recognition of different types of finger interactions

With the configuration of electrodes on the fingertips, we identify three finger interactions that can be recognized with HandSense. In this section, we illustrate the signal signature of each finger interaction with an example gesture. For better visualization, we also include the time boundary for each gesture in gray boxes.

a. Direct over-the-air electrode proximity detection. Fig. 5.8 we show the mean and standard deviation of the received signal at a receiving electrode when the transmitting and receiving electrodes are on the index finger and the thumb respectively, and two electrodes are kept in parallel at different distances. The received signal decreases exponentially with increase in distance between transmitting and receiving electrodes. We observe that beyond 5cm, the received signal stays at a minimum level, which is the capacitive coupling between signal traces on the processing board, thus electrode distances more than 5cm are hard to detect.

As an illustration, consider the closing fist gesture shown in Fig. 5.7(a), in which all the fingers are curled toward the palm to make a fist. The time series of 4 links of adjacent fingers (pinky-ring, ring-middle, middle-index, index-thumb) all show signal increase as the fingers in each pair move close toward each other.

Note that since the capacitive coupling amount depends not only on distance between electrodes, but also on orientation of electrodes to each other, as well as possible capacitive coupling to the user's hand, there is no direct mapping between received signal amplitude and electrode distance. However, as we are interested in *dynamic* finger gestures, the relative change in time in each data stream is the more important feature to recognize different gestures.

b. Detection of finger touching. When the two electrodes touch each other (i.e., the insulation over the signal electrodes), the capacitive coupling between the two would be the strongest. This strong capacitive coupling produces saturated readings at the output of the frontend receiver. An example of this finger interaction is the index pinch gesture (Fig. 5.7(b)), in which index finger tip taps on thumb tip. The index finger electrode acts as transmitter and the thumb electrode acts as receiver. The received signal at the thumb

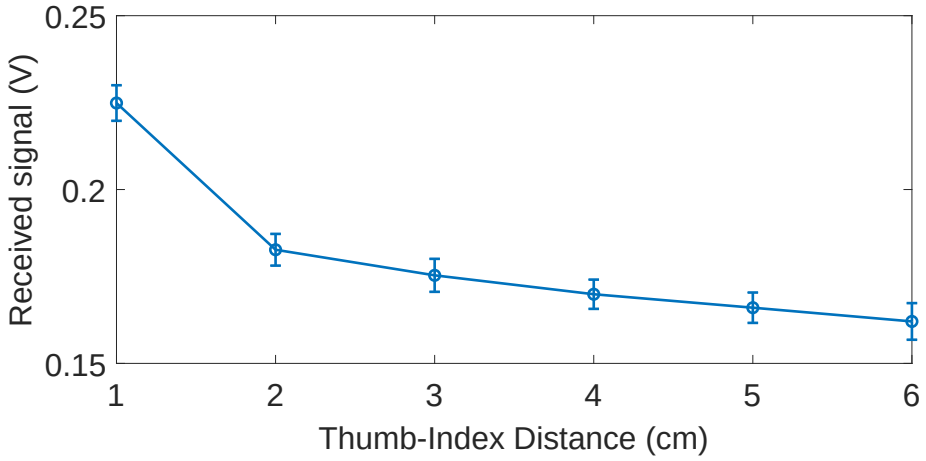


Figure 5.8: Received signal vs. distance (Thumb to index finger)

electrode quickly increases and saturates at 0.5V when the two fingers touch each other.

c. Indirect through-the-hand electrode communication. The short range (under 5cm) of the over-the-air electrode proximity detection makes the capacitive link between far apart fingers (e.g. thumb-to-pinky) seem unusable. However, we can take advantage of the palm as a communication channel between them. We discovered that when two electrodes are touching near the center of the palm at the same time, since the human hand is conductive, there is some capacitive coupling through the hand between the two electrodes. We can take advantage of this fact to use in some intuitive and low-effort finger gestures.

As an illustration, consider the tap gesture shown in Fig. 5.7(c), in which the thumb taps onto the surface of the index finger. The middle, ring, and little fingers are curled into the palm and thus electrodes on these fingers are coupled to the user's hand palm region. Fig. 5.7(c) also shows the time series signal on three channels, from thumb to middle, ring, and little fingers, when the user performs multiple tap gestures. As can be seen in this figure, when the thumb taps the base of the index fingers, received signals on all these three channels increase because of more capacitive coupling through the hand in each link. This provides features to differentiate this gesture in the classification step.

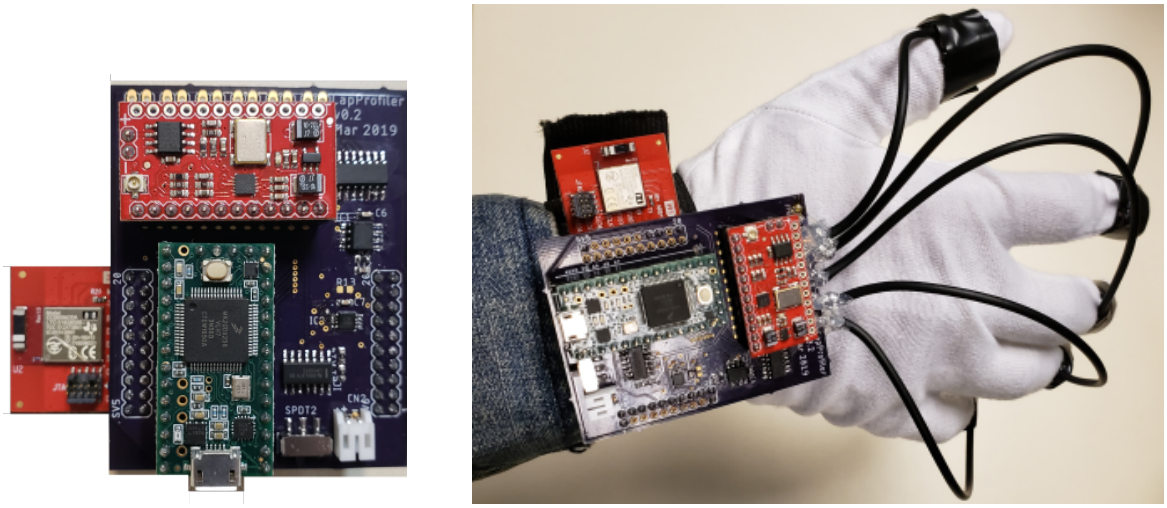
5.6.2 Neural network-based gesture classification

The input to HandSense's gesture classification system is a time series data of data samples, each contains received signal amplitudes in 10 links being calculated from the CapProfiler board. There are different approaches for the problem of time series classification [145]. Classicial machine learning techniques, such as SVM, Decision Tree, Logistic Regression, require manual feature extraction from the raw sensor data before feeding into their classifiers. However, handcrafted features have several challenges, such as task or application dependence, reliance on domain knowledge, difficulty in transferring to a new type of sensor data [146].

We instead employ a data-driven approach. In particular, we seek to train end-to-end models that allow raw sensor data as input data for gesture classification. We utilize several common neural network-based methods for Time Series Classification problems, in particular Multi Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Long Short-Term Memory Network (LSTM). The architectural details of each network is as follows.

Multi Layer Perceptron (MLP). As a baseline, we started with a simple neural network model as follows: Each input sequence is reshaped to a column vector of size 10 $\times$ [number of time steps]. The input layer is fully connected to a hidden layer, which is in turn fully connected to an output layer. The number of hidden neurons is set to approximately 2/3 $\times$ (number of input neurons + number of output neurons).

Convolutional Neural Network (CNN). CNN is used frequently with time series data problems, thanks to its ability to learn spatial/temporal relationship in the input data. In our experiment CNN network architecture, each input sequence is reshaped to a two-dimensional feature matrix, one dimension size is 10 (number of links being calculated), the other dimension is the time steps in the sequence. We pad input data with zeros to make input sequences of the same size. These input sequences are then used to train a Convolutional Neural Network (CNN). Our CNN consists of two convolutional layers, each followed by a max pooling layer. The kernel sizes for the convolution layers are 5 $\times$ 10 and 20 $\times$ 1. The pool sizes are 2 $\times$ 1 and 20 $\times$ 1. We use Rectified Linear (ReLU) activation



(a) CapProfiler board. (b) Glove prototype.

Figure 5.9: Prototype.

function after each convolutional layer and dropout of rate 0.4 after the fully connected layer. The initial learning rate is set at 10^{-3}.

Long Short-Term Memory network(LSTM). LSTM is a special kind of recurrent neural network (RNN) that is capable of learning long-term dependencies. Compared to standard feedforward neural networks (e.g. MLP and CNN) that feeds the whole sequence as an entire input, LSTM is able to learn the dependencies from time-series data by feeding the sequence to the network step by step. We experimented with a LSTM network with one hidden layer consisting of 50 LSTM units. Dropout layers and L2 regularization are used to avoid over-fitting the model. We set the initial learning rate to 10^{-3}.

5.7 Evaluation

In this section, we present our developed prototype, the set of dynamic, micro finger gestures used in our experiments, then describe the data collection process from users. Next, we evaluate the capability of HandSense in recognizing these gestures.

5.7.1 CapProfiler prototype

We designed a capacitive profiler board consisting of the following components: a Teensy 3.2 microcontroller [73] to do the central processing, a SparkFun Minigen signal generator [147], which is centered around the chip AD9837 [148], to generate sinusoidal wave, a custom analog receiver front-end circuit for displacement current measurements, two 8-channel CD74HC4051 multiplexers [149] for transmitting and receiving directions, and CC2650 BoosterPack [150] for Bluetooth data streaming. The signal generator generates a 1V peak-to-peak 100KHz signal. Fig. 5.9(a) shows the fabricated board.

We use a cotton glove and attach electrodes around its fingertips. The electrodes are connected to the central processing board by coaxial cables to avoid affect from environment noise (Fig. 5.9(b)).

5.7.2 Gesture set

HandSense is able to recognize different types of finger interactions as described above, giving us an opportunity to specify intuitive and low-effort finger gestures for operations on a head-mounted device. We design a set of such gestures, illustrated in Table 5.1. These gestures are highly suitable for operations on head-mounted devices, for example:

- Sliding (right to left or left to right): to rewind or fast forward any video
- Swiping: to scroll up or down a document
- Tap / double tap: to select an item on screen
- Closed fist: to close the current document / window
- Knob turn: to rotate displayed objects
- Pinch (between thumb and the remaining four fingers): to select between different options by pressing virtual buttons

The gestures in the set also demonstrate the capability of HandSense in recognizing the finger interactions described in Section 5.6. Also, most gestures require only small

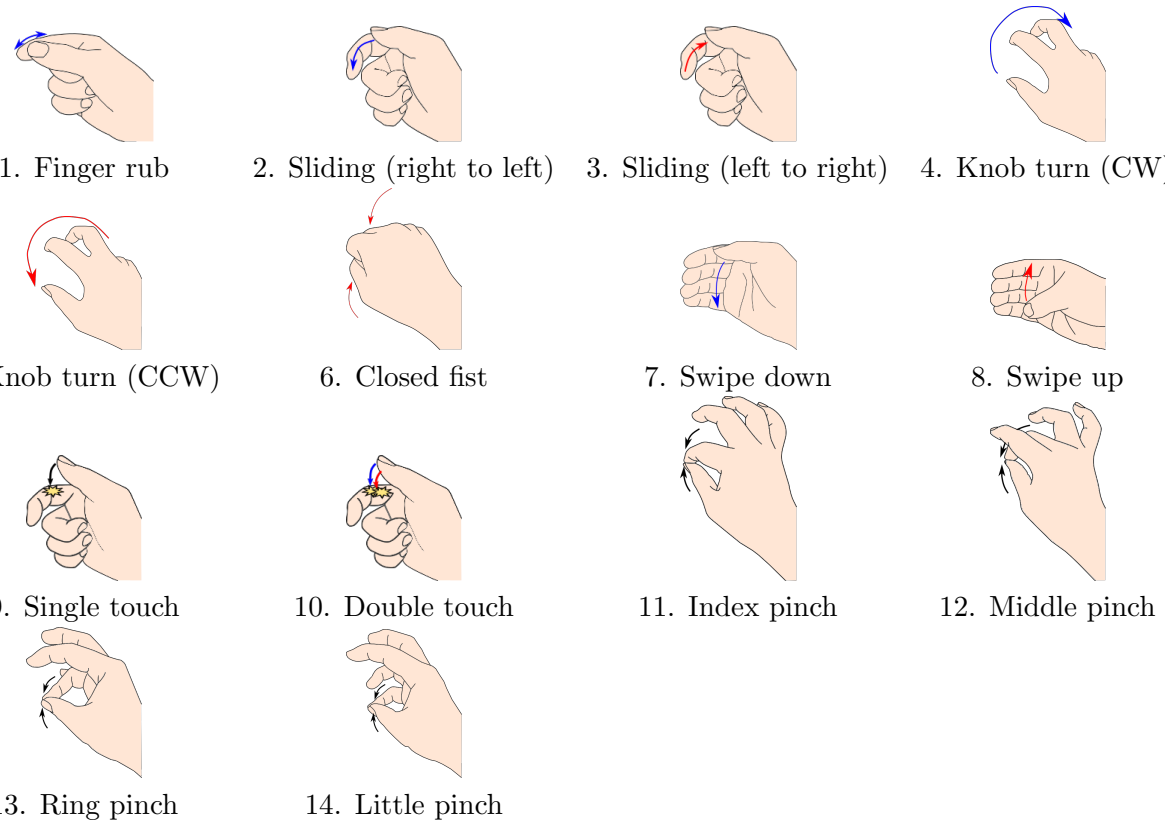


Table 5.1: Full gesture set used in our experiments. Note that the illustrations do not include the hand glove.

amount of motions and can be performed by muscles controlling the fingers, rather than those involving larger muscle groups. Note that the gesture set includes a few challenging pairs of gestures, such as sliding right to left vs. left to right, knob turn clockwise vs. counter-clockwise, which can be easily misclassified with each other.

5.7.3 Data collection and preprocessing

Ten subjects wore the glove on the right hand and performed the gestures; the glove is equipped with electrodes on fingertips, connected with the CapProfiler board worn on the subject's wrist, as described in above section. Each gesture is captured 25 times, with experiment sessions lasting about 30 minutes per user. To simplify analysis, the start and the end of each gesture are manually marked by pressing a button. In total, we captured $10 \times 25 \times 14 = 3500$ sequences with different lengths. This dataset is used in most of our experiments.

Based on the markers of the start and the end of each gesture, sequences are extracted into individual gesture windows. The time series data is further processed by a Hampel filter, followed by a moving average filter, before being given as input to the classification system.

5.7.4 Gesture recognition performance

We use common metrics for multi-class classification: precision, recall and F1-score, to evaluate the performance of each model in recognizing different finger gestures. We use 10-fold cross validation for evaluating these performance metrics. Table 5.2 shows these metrics for three models: Multi Layer Perceptron (MLP), Convolutional Neural Network (CNN) and Long Short-Term Memory network (LSTM).

Model	**Precision**	**Recall**	**F1 score**
MLP	0.909	0.903	0.903
CNN	0.945	0.942	0.942
LSTM	0.976	0.975	0.975

Table 5.2: Classification performance of different neural network-based methods.

MLP achieves 0.909 precision, 0.903 recall and 0.903 F1 score, which is a good baseline for the classification. This shows that with a simple fully connected layers model, the discriminative signatures in the capacitive profiling are already able to provide reasonably high accuracy in finger gesture classification. However, since MLP concatenates time series sequences on all the links into a single 1D input vector and samples are treated as independent neurons, it loses the temporal dependency within a single link as well as across links. For example, in swipe down gesture, the capacitive coupling amount should increase then decrease in this order: thumb-index, thumb-middle, thumb-ring, then thumb-little. As we will see, most of the misclassifications in MLP happen within close pairs of gestures: sliding right to left vs. left to right, knob turn clockwise vs. counter-clockwise, swiping up vs. down, single-touch vs. double-touch.

CNN performs better than MLP (0.945 precision, 0.942 recall and 0.943 F1 score). This is because CNN keeps the 2-dimensional data (10 links $\times$ number of time steps) as input to the network, and its 2D filters are able to learn the temporal dependency within each

link (e.g. received signal rises and falls when fingers move closer and further away) as well as across links (e.g. swipe down gesture, as described above).

LSTM has been widely recognized to achieve excellent performance on time series data classification. Compared to CNN, LSTM has better memorization of the long term dependencies of the past. Based on our results in Table 5.2, we show that LSTM achieves the best performance in all three metrics (0.976 precision, 0.975 recall and 0.975 F1 score).

Fig. 5.10 shows the confusion matrix of the finger gesture classification with the three neural network-based methods, using 10-fold cross validation. As expected, we can see that most misclassifications are within pairs of close gestures: sliding left to right vs. right to left, knob turn clockwise vs. counter-clockwise. CNN performs better than MLP in differentiating gestures in each pair, thanks to its awareness of temporal dependency in the data stream. LSTM further shows its superior performance on detection of compounding gestures such as single touch and double touch. Knob turn (clockwise, counter-clockwise) are the most challenging gestures for all three models. This can be for two reasons: 1) the highly similarity between signal traces of these two gestures, and 2) the gestures are hard to perform (feedback from users), meaning the collected signal might not have been consistent even for the same user.

Overall, the three neural network-based methods have high gesture recognition performance on our collected dataset, proving the distinctive signatures in the data stream collected from our measurement technique. Note that this exploration of neural network-based methods is by no means an exhaustive search for the model with the highest recognition performance. Instead, the focus is on the suitability of this new measured capacitive coupling profile in recognizing fine-grained finger gestures. The results in this section provide a promising baseline, and we leave additional analysis of suitable machine learning techniques for future work.

5.7.5 Microbenchmarks

Capacitive coupling measurement rate vs. classification accuracy. We evaluate the effect of the measurement rate on the classification accuracy of HandSense. From 100Hz-rate dataset collected from the above process, we downsampled the data stream to simulate data

collected at 50Hz, 25Hz, and 10Hz measurement rate. On these new datasets, we use the same CNN network architecture and 10-fold cross validation to evaluate the classification performance. Fig. 5.11 shows Precision, Recall, and F1 scores for these measurement rates. We can see that the classification performance degrades as the measurement rate decreases. This shows the advantage of our light-weight measurement technique in delivering high-rate measurements to classify fast, dynamic finger gestures more accurately.

Glove independency. Gloves used in HandSense system serve only as a convenient means to connect finger electrodes to the CapProfiler board on a wrist-worn device. To illustrate that the glove being used has little effect on the classification performance of HandSense, we asked one of the ten subjects above to wear a Hyper Tough Gripping Glove (Fig. 5.12) and collected another set of experiments from this subject. We then trained a CNN model using data collected from previous set of ten subjects when they wore the cotton glove, and tested this model on the newly collected data. The classification achieves 0.979 precision, 0.977 recall, and 0.977 F1 score, proving that training the HandSense classifier on only one glove allows the user to use other gloves as well.

5.8 Limitation and Discussion

Power consumption. For fast prototyping, our current CapProfiler prototype uses off-the-shelf modules, including a Teensy 3.2 microcontroller [73], a MiniGen signal generator module [147], and TI CC2650 BoosterPack for Bluetooth module [150]. At 3.7V supply voltage, the average current drawn in this unoptimized prototype is 90mA when Teensy is in active mode and 57mA when it is in sleep mode, with the breakdown for each component shown in Table 5.3. This means the CapProfiler board consumes 330mW in active mode and 211mW in sleep mode. While this is high power consumption, we believe power consumption can be reduced in an optimized prototype, given the simple functionalities of the CapProfiler board. Several power optimization methods can be: replacing MiniGen module with a simple microcontroller's pin toggle at the transmitted frequency, lowering measurement rate (increasing microcontroller's sleep time) while HandSense is in idle mode. We leave the power optimization of the CapProfiler board as the future work.

Component	Current drawn
Teensy (active mode)	38mA
Teensy (sleep mode)	5mA
Analog receiver frontend	2mA
CC2650 BoosterPack	10mA
MiniGen	40mA

Table 5.3: Current drawn in each component in our CapProfiler prototype.

Usability. Gloves are already prevalent in some workplace sites, such as repair and maintenance, and HandSense is easily adopted in these areas. While the current HandSense prototype remains bulky with coaxial cables connecting the finger electrodes with the CapProfiler board, given the minimal requirements for the glove (only finger electrodes and traces are needed), we believe it is possible to design cheap gloves with all sensing elements weaved into the fabric. Also, with the advance of skin electronics [130], the electrodes and traces can be attached directly to the user's hand, thus potentially enabling more applications of HandSense in consumer electronics.

Gesture spotting and gesture segmentation. Current system assumes well-defined start and end points of each gesture as the input to the classifier. We focused more on the sensor design and the suitability of measurement signal for the task of finger gesture classification. To be able to develop HandSense into a real-world system, other challenges still remain, such as detection of registered finger gestures versus random motion, segmentation of consecutive finger gestures, which we leave for future work.

Cross user training system. The performance metrics reported in previous section is for 10-fold cross validation, which simulates a per-person trained gesture classification system. We also experimented with the leave-one-person-out approach on the same dataset, and achieved lower performance (MLP: 0.682 Precision, 0.681 Recall, 0.649 F1 Score; CNN: 0.712 Precision, 0.701 Recall, 0.671 F1 Score; LSTM: 0.822 Precision, 0.815 Recall, 0.813 F1 Score). We believe with larger dataset, a more generalized model can be built to support cross-user training scenarios.

5.9 Related Work

Many finger gesture recognition and tracking methods have been proposed. We categorize them based on their sensing modalities.

Flex and inertial sensor based. Most early work such as Data Glove [128], Digital data entry glove [144] and CyberGlove [131] focused on sensing the amount of finger bending / flexing to infer hand and finger gestures. Bending / flexing the sensor would change its resistance in proportion to the amount of bending. Gloves like the AcceleGlove [142], are equipped with inertial sensors (accelerometer and gyroscope) which measure roll, pitch, yaw and provide absolute angular position to help reconstruct the exact posture of the hand. Accelerometers present in smartwatches [151] have also been leveraged to perform hand gesture recognition. Serendipity [124] recognizes up to 5 fine-grained finger gestures using inertial sensors inside smartwatches. Unfortunately, the sensors were expensive, heavy to carry, restricted free hand movement and usually required a user-specific calibration procedure.

Light and infrared based. Sayre Glove developed by Thomas de Fanti and Daniel Sandin [152] detects hand gestures based on the amount of light received at a photoreceiver. Optical fibers with an LED on one end and a photoreceiver on the other are connected along the fingers on the back of the hand. Bending a finger bends the optical fiber reducing the amount of received light. Other gloves like the MIT Glove [152] have LEDs stuck to the cloth and detectors track the light from these LEDs. More recently doing away with gloves, Aili [153] uses a table lamp and few low-cost photoreceivers to reconstruct a 3D hand skeleton in real time. ZeroTouch [154] makes use of infrared LEDs and sensors for hand pose sensing. While, ZeroTouch only tracks fingers in a 2D plane, Aili reconstructs 3D hand poses. SensIR [155] detects hand gestures with a wearable bracelet using infrared transmission and reflection. However, these solutions find it hard to translate small changes in flexion to micro or dynamic gestures. They also assume that the hands are always in the field of view of the sensor.

Magnetic field sensing. Magnetic tracking uses a source element radiating a magnetic field and a small sensor that reports its position and orientation with respect to the source.

They also do not rely on line-of-sight observation. Chouhan et al. [156] affix a strong magnet to the palm and Hall sensors on fingertips. When the fingers are brought close to the palm a low signal is sent to a microcontroller. uTrack [157] converts the thumb and fingers into a 3D input system using magnetic field sensing. A user wears a pair of magnetometers on the back of their fingers and a permanent magnet affixed to the back of the thumb. By moving the thumb across the fingers, a continuous location stream is obtained for 3D pointing. Similarly, Finexus [126] tracks precise motion of multiple fingertips by instrumenting the fingertips with electromagnets. These systems are often clunky and are heavily dependent on the range of magnetic field.

Acoustic sensing. Acoustic trackers use high-frequency sound to triangulate a source within the work area. Most systems send out pings from a source which are received by microphones in the environment. These systems rely on line-of-sight between the source and the microphones. The PowerGlove [158] uses two ultrasonic transmitters on the knuckles and a receiver on the TV, when a signal is received at the TV, triangulation is used to determine where the hand is in 3D space. FingerIO [123] transforms a device (typically a smart-phone) into an active sonar system that transmits inaudible sound signals and tracks the echoes from fingers at its microphones. FingerIO does not require instrumenting the finger with sensors and works even in the presence of occlusions between the finger and the device. Whereas, SoundTrak [159] requires users to wear a ring with an embedded miniaturize speaker sending an acoustic signal at a specific frequency which is captured by an array of miniature, inexpensive microphones on the target wearable device. The surface surrounding the transmitter and receiver greatly influences reflections.

WiFi and radar based. The basic idea is to leverage commercial off the shelf WiFi access points to transmit a signal and sense the effect of in-air hand motion on the wireless received signal strength, channel state information (CSI), angle of arrival. The performed hand gestures are mapped uniquely to changes in these values. WiGest [137] leverages changes in WiFi signal strength to sense in-air hand gestures around the users mobile device. Whereas, WiG [136] attempts to achieve a fine-grained gesture recognition only by observing abnormalities in CSI. Similarly, SignFi [160] recognizes upto 276 sign language hand gestures using CSI. WiDraw [161] harnesses the angle-of-arrival values of incoming wireless signals

at the mobile device to track the users hand trajectory. More recently, Google's Project Soli [122] proposed a 60GHz radar chip that is able to detect micro movements of fingers. Soli works on the principle of radar sensing, where using certain properties of the received back-scattered signal, hand gestures are inferred. Similar to light based sensors WiFi and radar based sensors require line of sight between transmitter and the hand.

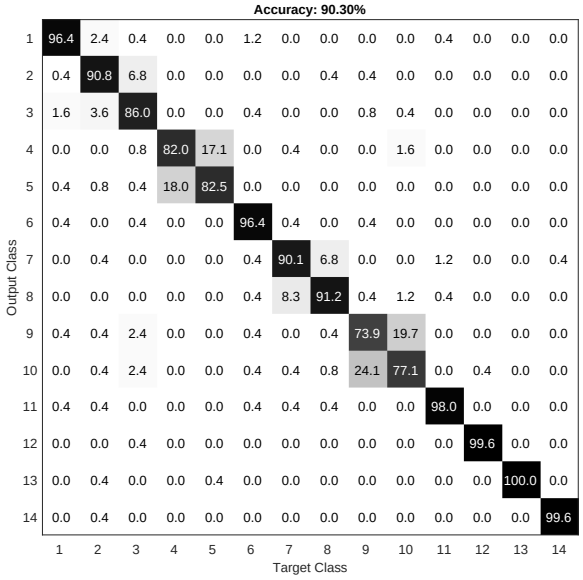
Camera and computer vision based. Multiple (depth) cameras capture raw images / video of the hand and the recorded raw data is then processed using sophisticated computer vision algorithms to determine the position and gesture being made. HoloLens [121] makes use of depth cameras present on the head mounted display to track hands in the field of view. Individual frames of the video are analyzed using algorithms to first separate the hands from the background, then detect the gesture from the image of the hand. DepthTouch [132] uses a depth-sensing camera, which reports a range value per pixel in addition to color, to track the 3D position of the users head and hand through a transparent vertical display screen. 6D Hands [133] uses two consumer-grade webcams to observe the users are hands. The pose made by the user is estimated by looking up the gesture from a precomputed database that relates hand silhouettes to their 3D configuration to enable more intuitive computer aided design (CAD). Keskin et al. [134] use the Kinect depth sensor to capture images. They then introduce a novel randomized decision forest (RDF) based hand shape classifier for pose estimation. Camera based techniques often raise privacy concerns and come with large computation overheads. They also require large datasets of gestures for reliable classification.

Electrical impedance tomography and capacitance based. Capacitance between two conductors depends on the distance and dielectric material between the conductors. GestureWrist [162] detects changes in wrist contour by measuring the capacitance between a series of electrodes integrated into a wristband. This is similar in operation to CapBand [2]. Electrical Impedance Tomography [163] is a similar paradigm employed for hand gesture recognition. Tomo [3] and Touche' [164] recover the inner impedance distribution of objects, forearm, wrist using pair-wised measurements from surface electrodes surrounding an object / forearm / wrist. Since the electrodes measure changes in object or muscle tension, the effort required to perform a gesture is high. Also these techniques find it harder to detect

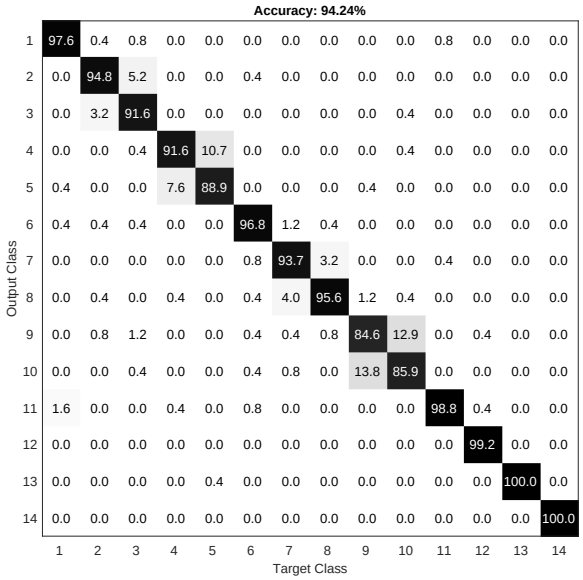
dynamic and fast gestures.

5.10 Conclusion

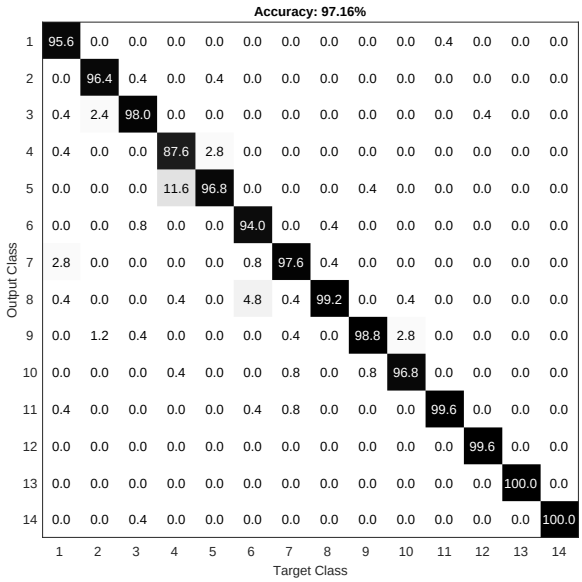
In this work, we introduce HandSense, a system based on pair-wise capacitive coupling measurements between electrodes placed on fingertips to recognize dynamic, micro finger gestures suitable for operations in Augmented Reality applications. We proposed a placement configuration for electrodes on the fingertips that minimizes the effect from the human hand to better associate the capacitive coupling measurements with inter-electrode distances. We designed a light-weight measurement technique based on synchronous undersampling to capture high-resolution capacitive profiling of fast, dynamic , micro finger gestures. The capacitive profiling is used in three end-to-end neural network-based models for gesture classification. Experiment results with our HandSense prototype show an average classification accuracy of 97% over a set of 14 dynamic, micro finger gestures from 10 different subjects. It achieves this accuracy without restrictions on hand position (as compared to cameras, for example) and with relatively lightweight instrumentation of the glove that enables use in environments where gloves are regularly changed. We believe our technique is a promising input interface to be used in conjunction with head-mounted augmented reality devices in working environments, which allows users to control the interface through finer gestures that are less interrupting to their workflow.



(a) MLP



(b) CNN



(c) LSTM

Figure 5.10: Confusion matrix of finger gesture classification using three neural network-based models.

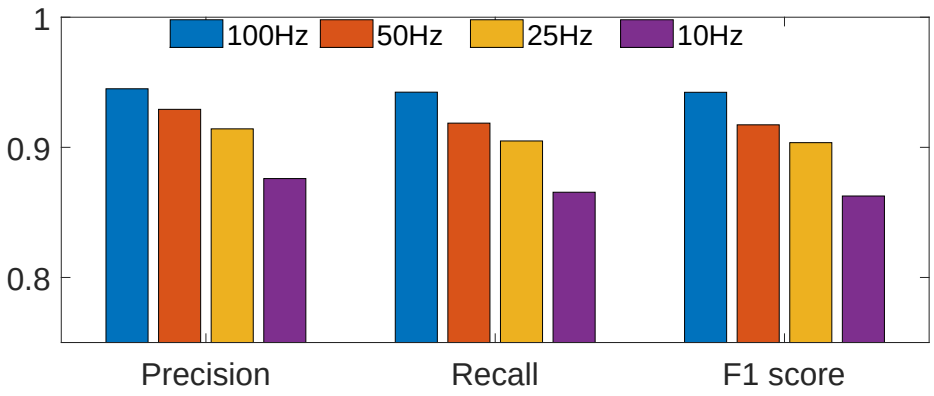


Figure 5.11: Effect of the measurement rate on classification performance.

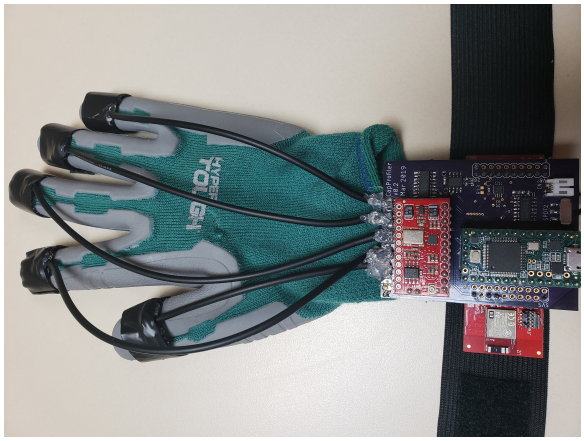


Figure 5.12: Different glove.

Chapter 6

Conclusion

In this dissertation, we have presented communication and sensing methods that enable seamless interactions between humans and their environments. These are based on two modalities, visible light and capacitive sensing, which are ubiquitous in indoor environments and on human body. Our approaches aim for minimal instrumentation both in the environments and on the users to ensure unobtrusive experience for the users. The communication techniques are well aligned with user's intentions, while the sensing techniques have low complexity. These communication and sensing methods are all demonstrated by systems built end-to-end, including designing, prototyping and evaluation. Overall, they present a holistic effort to make interactions between humans and their environments more intuitive, less obtrusive and less effort.

6.1 Summary of contributions

This dissertation has made the following contributions:

- We proposed a spatial content-adaptive encoding method in screen-to-camera communication to increase the goodput of the communication channel while maintaining normal viewing experience (flicker-free) for users. This technique would be helpful for users to quickly obtain side information from many screens available in public spaces by using built-in cameras inside their wearable glasses.

- We developed a secure yet convenient method for user identification, authorization and authentication when users interact with surrounding devices and objects. In particular, the technique aims to do authentication on every single user touch. It is based on a hardware token worn on user's body, such as a wristband, which interacts

with a receiver embedded inside the object through a body-guided channel established when the user touches the object. This technique has superior resilience to attacks, and robust authentication capability.

- We developed a system embedded in indoor lighting environment to sense the human occupancy and room activities. In this system, photosensors are integrated inside light bulbs to sense the light reflected off the floor. Light change and shadow caused by human activities inside the room are used to infer useful information, including localization, room occupancy level estimation, and room activity recognition.

- Lastly, for applications of Augmented Reality head-mounted devices in several industries, we proposed an always-available, on-hand, and light-weight system to recognize a series of dynamic, micro finger gestures that are suitable for controlling these head-mounted devices. In this system, electrodes are placed on fingertips to enable measurements of capacitive coupling between each pair of fingers. The capacitive profile of pair-wise capacitive coupling measurements would be used for a classification system to recognize different low-effort finger gestures.

6.2 Looking ahead

With devices becoming increasingly smaller, more capable, and more integrated into everyday objects, and communication systems becoming more reliable, ubiquitous computing would quickly be seen in more applications in our life. This thesis touched on several aspects to push ubiquitous computing towards wider adoption: more intuitive human-machine interfaces, designs with built-in security and privacy primitives, minimal instrumentation on human body and environments. While other challenges remain, including ones from technical, economic and social perspectives, we believe the communication and sensing techniques, as well as systems and devices, presented in this thesis would be ready for use in future Ubiquitous Computing applications.

References

[1] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, pp. 3–11, July 1999.

[2] H. Truong, S. Zhang, U. Muncuk, P. Nguyen, N. Bui, A. Nguyen, Q. Lv, K. Chowdhury, T. Dinh, and T. Vu, "Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pp. 54–67, ACM, 2018.

[3] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 167–173, ACM, 2015.

[4] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison, "Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, (New York, NY, USA), pp. 157–166, ACM, 2015.

[5] S. Shen, H. Wang, and R. Roy Choudhury, "I am a smartwatch and i can track my user's arm," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, (New York, NY, USA), pp. 85–96, ACM, 2016.

[6] A. Parate and D. Ganesan, *Detecting Eating and Smoking Behaviors Using Smartwatches*, pp. 175–201. Cham: Springer International Publishing, 2017.

[7] C. Karatas, L. Liu, H. Li, J. Liu, Y. Wang, S. Tan, J. Yang, Y. Chen, M. Gruteser, and R. Martin, "Leveraging wearables for steering and driver tracking," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, April 2016.

[8] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann, "Skeleton-based motion capture for robust reconstruction of human motion," in *Computer Animation 2000. Proceedings*, pp. 77–83, IEEE, 2000.

[9] N. R. Howe, M. E. Leventon, and W. T. Freeman, "Bayesian reconstruction of 3d human motion from single-camera video," in *Advances in neural information processing systems*, pp. 820–826, 2000.

[10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, ACM, 2011.

[11] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proc. of the 20th Annual Int. Conf. on Mobile Computing and Networking*, pp. 617–628, ACM, 2014.

[12] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity wi-fi," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '17, (New York, NY, USA), pp. 6:1–6:10, ACM, 2017.

[13] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI'14, (Berkeley, CA, USA), pp. 317–329, USENIX Association, 2014.

[14] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *Proc. of the 14th Annu. Int. Conf. on Mobile Syst., Applicat., and Services*, MobiSys '16, (New York, NY, USA), pp. 71–84, ACM, 2016.

[15] V. Nguyen, Y. Tang, A. Ashok, M. Gruteser, K. Dana, W. Hu, E. Wengrowski, and N. Mandayam, "High-rate flicker-free screen-camera communication with spatially adaptive embedding," in *IEEE INFOCOM 2016*, pp. 1–9, April 2016.

[16] V. Nguyen, M. Ibrahim, H. Truong, P. Nguyen, M. Gruteser, R. Howard, and T. Vu, "Body-guided communications: A low-power, highly-confined primitive to track and secure every touch," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), pp. 353–368, ACM, 2018.

[17] V. Nguyen, M. Ibrahim, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, "Eyelight: Light-and-shadow-based occupancy estimation and room activity recognition," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 351–359, April 2018.

[18] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng, "Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication," in *Proc. of the 13th Annual Int. Conf. on Mobile Systems, Applications, and Services (Mobisys)*, (New York, NY, USA), pp. 181–195, ACM, 2015.

[19] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proc. of the 13th Annual Int. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, (New York, NY, USA), pp. 197–211, ACM, 2015.

[20] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Review: Digital image steganography: Survey and analysis of current methods," *Signal Process.*, vol. 90, pp. 727–752, Mar. 2010.

[21] N. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, pp. 26–34, Feb 1998.

[22] B. Andrén, K. Wang, and K. Brunnström, "Characterizations of 3d tv: active vs passive," in *SID Symp. digest of technical papers*, vol. 43, pp. 137–140, 2012.

[23] H. DeLange, "Experiments of flicker and some calculations of an electrical analogue of the foveal systems.," *Physica*, 1952.

[24] D. Kelly, "Sine waves and flicker fusion," *Documenta Ophthalmologica*, vol. 18, no. 1, pp. 16–35, 1964.

[25] E. H. Adelson, "Lightness perception and lightness illusions," *New Cogn. Neurosci*, vol. 339, 2000.

[26] T. Cornsweet, *Visual perception*. Academic press, 2012.

[27] C. Chubb, G. Sperling, and J. A. Solomon, "Texture interactions determine perceived contrast," *Proceedings of the National Academy of Sciences*, vol. 86, no. 23, pp. 9631–9635, 1989.

[28] J. Davis, Y.-H. Hsieh, and H.-C. Lee, "Humans perceive flicker artifacts at 500 hz," *Scientific reports*, vol. 5, 2015.

[29] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Comput. Vision*, vol. 43, pp. 29–44, June 2001.

[30] O. G. Cula and K. J. Dana, "Recognition methods for 3d textured surfaces," in *Proceedings of SPIE Conf. on human vision and electronic imaging VI*, vol. 4299, p. 3, 2001.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Anal. Mach. Intell., IEEE Trans. on*, vol. 34, pp. 2274–2282, Nov 2012.

[32] `https://code.google.com/p/glvideoplayer/`.

[33] `http://ls.wim.uni-mannheim.de/de/pi4/research/projects/retargeting/test-sequences/`.

[34] `http://www.elementaltechnologies.com/resources/4k-test-sequences`.

[35] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Capacity of pervasive camera based communication under perspective distortions," in *Pervasive Computing and Communications (PerCom), 2014 IEEE Int. Conf. on*, pp. 112–120, March 2014.

[36] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: Interference-free wireless links using lcd-camera pairs," in *Proc. of the 16th Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*, (New York, NY, USA), pp. 137–148, ACM, 2010.

[37] T. Hao, R. Zhou, and G. Xing, "Cobra: Color barcode streaming for smartphone systems," in *Proc. of the 10th Int. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, (New York, NY, USA), pp. 85–98, ACM, 2012.

[38] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *Proc. of the 19th Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*, (New York, NY, USA), pp. 15–26, ACM, 2013.

[39] W. Hu, J. Mao, Z. Huang, Y. Xue, J. She, K. Bian, and G. Shen, "Strata: Layered coding for scalable visual communication," in *Proc. of the 20th Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*, (New York, NY, USA), pp. 79–90, ACM, 2014.

[40] A. Wang, S. Ma, C. Hu, J. Huai, C. Peng, and G. Shen, "Enhancing reliability to boost the throughput over screen-camera links," in *Proc. of the 20th Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*, (New York, NY, USA), pp. 41–52, ACM, 2014.

[41] W. Yuan, K. Dana, A. Ashok, M. Gruteser, and N. Mandayam, "Dynamic and invisible messaging for visual mimo," in *Proc. of the 2012 IEEE Workshop on the Applications of Computer Vision*, (Washington, DC, USA), pp. 345–352, IEEE Computer Society, 2012.

[42] G. Woo, A. Lippman, and R. Raskar, "Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter," in *Mixed and Augmented Reality, 2012 IEEE Int. Symp. on*, pp. 59–64, Nov 2012.

[43] "Amazon dash button." https://www.amazon.com/ddb/learn-more.

[44] C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov, "Using frankencerts for automated adversarial testing of certificate validation in ssl/tls implementations," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP '14, (Washington, DC, USA), pp. 114–129, IEEE Computer Society, 2014.

[45] R. Verdult, F. D. Garcia, and B. Ege, "Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer," in *Supplement to the Proceedings of 22nd USENIX Security Symposium (Supplement to USENIX Security 15)*, (Washington, D.C.), pp. 703–718, USENIX Association, 2015.

[46] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Network and Distributed System Security Symposium (NDSS) (to appear)*, 2011.

[47] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling, "Distinguishing users with capacitive touch communication," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, (New York, NY, USA), pp. 197–208, ACM, 2012.

[48] M. Hessar, V. Iyer, and S. Gollakota, "Enabling on-body transmissions with commodity devices," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, (New York, NY, USA), pp. 1100–1111, ACM, 2016.

[49] C. J. Yang and A. P. Sample, "Em-comm: Touch-based communication via modulated electromagnetic emissions," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 118:1–118:24, Sept. 2017.

[50] S. j. Song, S. J. Lee, N. Cho, and H. j. Yoo, "Low power wearable audio player using human body communications," in *2006 10th IEEE International Symposium on Wearable Computers*, pp. 125–126, Oct 2006.

[51] C. Holz and M. Knaust, "Biometric touch sensing: Seamlessly augmenting each touch with continuous authentication," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, (New York, NY, USA), pp. 303–312, ACM, 2015.

[52] K. Partridge, B. Dahlquist, A. Veiseh, A. Cain, A. Foreman, J. Goldberg, and G. Borriello, "Empirical measurements of intrabody communication performance under varied physical configurations," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 183–190, ACM, 2001.

[53] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, pp. 1727–1765, Sept 2016.

[54] "Comparing Low-Power Wireless Technologies." `https://goo.gl/sYPVzM`.

[55] M. Ghamari, H. Arora, R. S. Sherratt, and W. Harwin, "Comparison of low-power wireless communication technologies for wearable health-monitoring applications," in *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 1–6, 2015.

[56] "Antenna Circuit Design for RFID Applications." `http://ww1.microchip.com/downloads/en/AppNotes/00710c.pdf`.

[57] T. P. Diakos, "Eavesdropping near-field contactless payments: a quantitative analysis," *The Journal of Engineering*, vol. 2013, pp. 48–54(6), October 2013.

[58] N. Roy and R. R. Choudhury, "Ripple II: Faster communication through physical vibration," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, (Santa Clara, CA), pp. 671–684, USENIX Association, 2016.

[59] J. Adkins, G. Flaspohler, and P. Dutta, "Ving: Bootstrapping the desktop area network with a vibratory ping," in *Proceedings of the 2Nd International Workshop on Hot Topics in Wireless*, HotWireless '15, (New York, NY, USA), pp. 21–25, ACM, 2015.

[60] "LRA." `goo.gl/sBYDLH`.

[61] "Microchip BodyCom Technology." `http://ww1.microchip.com/downloads/en/DeviceDoc/30685a.pdf`.

[62] M. D. Pereira, G. A. Alvarez-Botero, and F. R. de Sousa, "Characterization and modeling of the capacitive hbc channel," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, pp. 2626–2635, Oct 2015.

[63] J. Bae and H. J. Yoo, "The effects of electrode configuration on body channel communication based on analysis of vertical and horizontal electric dipoles," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, pp. 1409–1420, April 2015.

[64] M. Seyedi, B. Kibret, D. T. H. Lai, and M. Faulkner, "A survey on intrabody communications for body area network applications," *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 2067–2079, Aug 2013.

[65] B. Kibret, M. Seyedi, D. T. H. Lai, and M. Faulkner, "Investigation of galvanic-coupled intrabody communication using the human body circuit model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1196–1206, July 2014.

[66] M. Seyedi, B. Kibret, D. T. H. Lai, and M. Faulkner, "A survey on intrabody communications for body area network applications," *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 2067–2079, Aug 2013.

[67] M. A. Callejon, D. Naranjo-Hernandez, J. Reina-Tosina, and L. M. Roa, "A comprehensive study into intrabody communication measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, pp. 2446–2455, Sept 2013.

[68] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, "Applying electric field sensing to human-computer interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 280–287, ACM Press/Addison-Wesley Publishing Co., 1995.

[69] T. G. Zimmerman, "Personal area networks: Near-field intrabody communication," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 609–617, 1996.

[70] "Fundamentals of Electrostatic Discharge." `https://goo.gl/y5UEwG`.

[71] "Time-Based One-Time Password Algorithm." `https://tools.ietf.org/html/rfc6238`.

[72] S. Golomb and R. Scholtz, "Generalized barker sequences," *IEEE Transactions on Information Theory*, vol. 11, pp. 533–537, October 1965.

[73] "Teensy 3.2 board." `https://goo.gl/Qt5tYt`.

[74] "INA332." `http://www.ti.com/product/INA332`.

[75] "AD835." `http://www.analog.com/en/products/linear-products/analog-multipliers-dividers/ad835.html`.

[76] "Analog Discovery 2." `https://goo.gl/sbfwSw`.

[77] "LT1563." `http://www.linear.com/product/LTC1563`.

[78] "MSP432 Launchpad." `https://goo.gl/vucGRm`.

[79] H. Truong, P. Nguyen, V. Nguyen, M. Ibrahim, R. Howard, M. Gruteser, and T. Vu, "Through-body capacitive touch communication," in *Proceedings of the 9th ACM Workshop on Wireless of the Students, by the Students, and for the Students*, S3 '17, (New York, NY, USA), pp. 7–9, ACM, 2017.

[80] H. Truong, P. Nguyen, A. Nguyen, N. Bui, and T. Vu, "Capacitive sensing 3d-printed wristband for enriched hand gesture recognition," in *Proceedings of the 2017 Workshop on Wearable Systems and Applications*, WearSys '17, (New York, NY, USA), pp. 11–15, ACM, 2017.

[81] "Amazon IoT button." `https://aws.amazon.com/iotbutton/`.

[82] "MSP430G2553." `http://www.ti.com/product/MSP430G2553`.

[83] F. Schaub, R. Deyhle, and M. Weber, "Password entry usability and shoulder surfing susceptibility on different smartphone platforms," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, (New York, NY, USA), pp. 13:1–13:10, ACM, 2012.

[84] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, (Berkeley, CA, USA), pp. 1–7, USENIX Association, 2010.

[85] P. Nguyen, U. Muncuk, A. Ashok, K. R. Chowdhury, M. Gruteser, and T. Vu, "Battery-free identification token for touch sensing devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, (New York, NY, USA), pp. 109–122, ACM, 2016.

[86] A. De Luca, A. Hang, E. von Zezschwitz, and H. Hussmann, "I feel like i'm taking selfies all day!: Towards understanding biometric authentication on smartphones," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), pp. 1411–1414, ACM, 2015.

[87] "Apple face security." https://goo.gl/XvP2Wu.

[88] V. Roth, P. Schmidt, and B. Güldenring, "The ir ring: Authenticating users' touches on a multi-touch display," in *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, (New York, NY, USA), pp. 259–262, ACM, 2010.

[89] A. Bianchi and S. Je, "Disambiguating touch with a smart-ring," in *Proceedings of the 8th Augmented Human International Conference*, AH '17, (New York, NY, USA), pp. 27:1–27:5, ACM, 2017.

[90] J. Yun and S.-S. Lee, "Human movement detection and identification using pyroelectric infrared sensors," *Sensors*, vol. 14, no. 5, pp. 8057–8081, 2014.

[91] C. R. Wren and E. M. Tapia, "Toward scalable activity recognition for sensor networks," in *LoCA*, vol. 3987, pp. 168–185, 2006.

[92] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using rgb-d," UbiComp '12, pp. 208–211, ACM, 2012.

[93] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "Pbn: Towards practical activity recognition using smartphone-based body sensor networks," in *Proc. of the 9th ACM Conf. on Embedded Networked Sensor Syst.*, pp. 246–259, ACM, 2011.

[94] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, (Seattle, WA).

[95] C. Zhang and X. Zhang, "Litell: Robust indoor localization using unmodified light fixtures," in *Proc. of the 22nd Annu. Int.Conf. on Mobile Computing and Networking*, MobiCom '16, (New York, NY, USA), pp. 230–242, ACM, 2016.

[96] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proc. of the 21st Annu. Int. Conf. on Mobile Computing and Networking*, MobiCom '15, (New York, NY, USA), 2015.

[97] Y. Yang, J. Hao, J. Luo, and S. J. Pan, "Ceilingsee: Device-free occupancy inference through lighting infrastructure based led sensing," in *Proc. of the 15th Int. Conf. on Pervasive Computing and Commun.*, Percom '17, IEEE, 2016.

[98] M. Ibrahim, V. Nguyen, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, "Visible light based activity sensing using ceiling photosensors," in *Proc. of the 3rd Workshop on Visible Light Commun. Syst.*, VLCS '16, (New York, NY, USA), pp. 43–48, ACM, 2016.

[99] `http://www.energy.gov/energysaver/led-lighting`.

[100] `http://www.ledsmagazine.com/articles/2005/01/benefits-and-drawbacks-of-leds.html`.

[101] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proc. IEEE INFOCOM 2000 Conf. on Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. and Commun. Soc.*, vol. 2, pp. 775–784 vol.2, 2000.

[102] M. Ibrahim, M.and Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Trans. on Vehicular Technology*, 2012.

[103] Z. K. F. Adib and D. Katabi, "Multi-person localization via rf body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI'15)*, (Oakland, CA).

[104] K. E. Caine, A. D. Fisk, and W. A. Rogers, "Benefits and privacy concerns of a home equipped with a visual sensing system: A perspective from older adults," in *Proc. of the human factors and ergonomics society annual meeting*, vol. 50, 2006.

[105] M. Valtonen, J. Maentausta, and J. Vanhala, "Tiletrack: Capacitive human tracking using floor tiles," in *2009 IEEE Int. Conf. on Pervasive Computing and Commun.*, pp. 1–10, March 2009.

[106] R. J. Orr and G. D. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *CHI '00 Extended Abstracts on Human Factors in Computing Syst.*, CHI EA '00, (New York, NY, USA), pp. 275–276, ACM, 2000.

[107] `http://ab.rockwellautomation.com/Sensors-Switches/Operator-Safety/Light-Curtain`.

[108] `https://www.pepperl-fuchs.com/global/en/classid_4294.htm`.

[109] Z. Li, W. Chen, C. Li, M. Li, X.-Y. Li, and Y. Liu, "Flight: Clock calibration using fluorescent lighting," in *Proc. of the 18th Annu. Int. Conf. on Mobile Computing and Networking*, Mobicom '12, (New York, NY, USA), pp. 329–340, ACM, 2012.

[110] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Pick, and W. H. Warren, "Visual flow influences gait transition speed and preferred walking speed," *Experimental brain research*, vol. 181, no. 2, pp. 221–228, 2007.

[111] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.

[112] `http://www.ti.com/product/LF356`.

[113] `http://www.ti.com/lit/ds/symlink/ina126.pdf`.

[114] `http://www.ti.com/product/MSP432P401R`.

[115] `http://www.ti.com/tool/cc3100boost`.

[116] `https://www.stereolabs.com/`.

[117] "Philips Azurion." `https://www.usa.philips.com/healthcare/resources/landing/azurion`.

[118] "Trimble Mixed Reality." `https://mixedreality.trimble.com/`.

[119] "A Use Case for Digital Field Service With the Microsoft HoloLens." `https://www.sikich.com/insight/microsoft-hololens-use-case-for-digital-field-service/`.

[120] "Packing with Mixed Reality: KLM uses Microsoft HoloLens to redefine its cargo training experience with mixed reality." `https://customers.microsoft.com/en-gb/story/klm-airlines-travel-and-transportation-hololense-azure-netherlands`.

[121] "Microsoft HoloLens 2." `https://www.microsoft.com/en-us/hololens`.

[122] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), pp. 851–860, ACM, 2016.

[123] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), pp. 1515–1525, ACM, 2016.

[124] H. Wen, J. Ramos Rojas, and A. K. Dey, "Serendipity: Finger gesture recognition using an off-the-shelf smartwatch," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), pp. 3847–3851, ACM, 2016.

[125] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, (New York, NY, USA), pp. 167–173, ACM, 2015.

[126] K.-Y. Chen, S. N. Patel, and S. Keller, "Finexus: Tracking precise motions of multiple fingertips using magnetic sensing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), pp. 1504–1514, ACM, 2016.

[127] T. Li, X. Xiong, Y. Xie, G. Hito, X.-D. Yang, and X. Zhou, "Reconstructing hand poses using visible light," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 71:1–71:20, Sept. 2017.

[128] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A hand gesture interface device," in *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, (New York, NY, USA), pp. 189–192, ACM, 1987.

[129] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 461–482, 2008.

[130] "The Future of Skin Electronics." `https://www.youtube.com/watch?v=zpGujcLRHNw`.

[131] "Cyberglove ii." http://www.cyberglovesystems.com/cyberglove-ii.

[132] H. Benko and A. D. Wilson, "Depthtouch: using depthsensing camera to enable freehand interactions on and above the interactive surface," in *In Proceedings of the IEEE Workshop on Tabletops and Interactive Surfaces*, Citeseer, 2009.

[133] R. Wang, S. Paris, and J. Popović, "6d hands: markerless hand-tracking for computer aided design," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 549–558, ACM, 2011.

[134] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *European Conference on Computer Vision*, pp. 852–863, Springer, 2012.

[135] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[136] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, IEEE, 2015.

[137] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1472–1480, IEEE, 2015.

[138] J. Smith, *Electric Field Imaging*. PhD thesis, Massachusetts Institute of Technology, 1999.

[139] "Worker deaths by electrocution." https://www.cdc.gov/niosh/docs/98-131/pdfs/98-131.pdf.

[140] S. Banerjee and M. Levy, "Approximate capacitance expressions for two equal sized conducting spheres," *Proc. ESA Annu. Meet. Electrostatics*, 2014.

[141] "Synchronous sampling and algorithmic amplitude detection." `https://www.eetimes.com/document.asp?doc_id=1253691#`.

[142] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman, "The acceleglove: A whole-hand input device for virtual reality," in *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, (New York, NY, USA), pp. 259–259, ACM, 2002.

[143] H. G. Kortier, V. I. Sluiter, D. Roetenberg, and P. H. Veltink, "Assessment of hand kinematics using inertial and magnetic sensors," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 70, 2014.

[144] G. Grimes, "Digital data entry glove interface device." https://patents.google.com/patent/US4414537A/en, (1981).

[145] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, Mar 2019.

[146] H. Nweke, T. Wah, M. Ali Al-garadi, and U. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, 04 2018.

[147] "SparkFun MiniGen." `https://www.sparkfun.com/products/11420`.

[148] "Analog devices ad9837 datasheet." https://www.analog.com/media/en/technical-documentation/data-sheets/AD9837.PDF.

[149] "Texas instruments cd74hc4051 datasheet." http://www.ti.com/lit/ds/symlink/cd74hc4051-ep.pdf.

[150] "TI SimpleLink Bluetooth low energy CC2650 Module BoosterPack Plug-in Module." `http://www.ti.com/tool/BOOSTXL-CC2650MA`.

[151] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 9–14, ACM, 2015.

[152] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer graphics and Applications*, vol. 14, no. 1, pp. 30–39, 1994.

[153] T. Li, X. Xiong, Y. Xie, G. Hito, X.-D. Yang, and X. Zhou, "Reconstructing hand poses using visible light," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 71, 2017.

[154] J. Moeller and A. Kerne, "Zerotouch: an optical multi-touch and free-air interaction architecture," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2165–2174, ACM, 2012.

[155] J. McIntosh, A. Marzo, and M. Fraser, "Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, (New York, NY, USA), pp. 593–597, ACM, 2017.

[156] T. Chouhan, A. Panse, A. K. Voona, and S. Sameer, "Smart glove with gesture recognition ability for the hearing and speech impaired," in *2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS)*, pp. 105–110, IEEE, 2014.

[157] K.-Y. Chen, K. Lyons, S. White, and S. Patel, "utrack: 3d input using two magnetic sensors," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 237–244, ACM, 2013.

[158] Mattel, "Power glove." https://www.microsoft.com/buxtoncollection/a/pdf/PowerGlove(1989).

[159] C. Zhang, Q. Xue, A. Waghmare, S. Jain, Y. Pu, S. Hersek, K. Lyons, K. A. Cunefare, O. T. Inan, and G. D. Abowd, "Soundtrak: Continuous 3d tracking of a finger using active acoustics," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 30:1–30:25, June 2017.

[160] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, pp. 23:1–23:21, Mar. 2018.

[161] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 77–89, ACM, 2015.

[162] J. Rekimoto, "Gesturewrist and gesturepad: Unobtrusive wearable interaction devices," in *Proceedings Fifth International Symposium on Wearable Computers*, pp. 21–27, IEEE, 2001.

[163] Y. Zhang, R. Xiao, and C. Harrison, "Advancing hand gesture recognition with high resolution electrical impedance tomography," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), pp. 843–850, ACM, 2016.

[164] M. Sato, I. Poupyrev, and C. Harrison, "Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 483–492, ACM, 2012.