Towards Subtle and Continuously Available Input Devices for the Modern Wearable Devices

Farshid Salemi Parizi

A dissertation submitted in partial fulfillment of the
requirement for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Shwetak N. Patel, Chair

Joshua R. Smith

Jon Froehlich

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Towards Subtle and Continuously Available Input Devices for the Modern Wearable Devices

Farshid Salemi Parizi

Chair of the Supervisory Committee:
Washington Research Foundation Entrepreneurship Endowed Professor Shwetak N. Patel
Paul G. Allen School of Computer Science and Engineering & Department of Electrical and Computer Engineering

Wearable computing platforms, such as smartwatches and head-mounted mixed reality displays represent a promising direction for next-generation computing platforms. As computing platforms evolve from devices we carry with us to devices we wear on us, there is a renewed demand for input techniques that are decoupled from the display. These input devices demand precise tracking for high-fidelity interaction and must be continuously available without requiring extensive calibrations and subtle to support various contexts of use while enabling robust and expressive interaction. In this thesis, I focus on enabling new input devices that power more natural and robust handheld controllers and wearable devices that support subtle finger and wrist-based interactions through innovations in both the hardware design and tracking algorithm.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who have helped me to reach this significant milestone of my life. First, my family who without all their sacrifice, I would have not got here. They have influenced me tremendously and I will always remember and appreciate all their hard work to help me thrive. Mom and dad, your constant love and support is the most important ingredient in all my achievements. My sister, Mahshid; you made my life so much easier here in Seattle. I am so happy and thankful you were here to help me in so many situations.

Second, I want to express sincere appreciation to Prof. Shwetak Patel for being an amazingly talented and supportive advisor. You basically have eliminated all the logistical hardship graduate students go through, and you are the main reasons which made my graduate studies more enjoyable that many other students. Shwetak, you gave me the freedom to work on whatever projects I wanted to and I never had to worry about funding. I'll never take these luxuries for granted. Your creativity, optimism, generosity, and energy will always remain inspirational and I feel very lucky to be working with you. Additionally, I also like to thank Prof. Josh Smith who allowed me to work with him as my first big project during my undergraduate studies. I also like to thanks all the amazing feedback you have provided me throughout my PhD. I also want to thank the many industry mentors I've have the privilege to work with over the years. Wolf Kienzle, Hrvoje Benko, Aakar Gupta, Teddy Seyed, Steve Hodges, and Andy Wilson. Learning from all of you has given me a valuable breadth of perspectives and I am forever thankful for the opportunities you gave me. Lastly, I like to thank Saman Naderiparizi. You were a great mentor, not only for my graduate studies but important life lessons as well.

Third I would like to thank all the members of the Ubicomp Lab at UW for all their help, discussion, and guidance. Alex Mariakakis, thanks so much for the numerous times you helped me with deadlines, presentations, goals, English language and basically many, many questions that I asked you during my PhD. I also like to thank my colleagues from the lab Josh Fromm, Anandghan Waghmare, Ishan Chatterjee, Yuntao Wang, Shirley Xue, Jason Hoffman, Alvin Cao, and Hanchuan Li. It was great working with you all and I learned so much from you all in the process. Lastly, I want to greatly thank Eric Whitmire. I could have not asked for a better mentor, collaborator and friend. I enjoyed working with you on many projects and you have completely changed the way I think, work and approach problems. I am forever thankful to you for opening so many opportunities to me.

DEDICATION

to my dear parents, Faranges & Abbas who I love.

Chapter 1

INTRODUCTION

1.1 Motivation

Since the invention of the first digital computers, not only computers are becoming more powerful, and smaller but also they are now more tightly coupled to our body which has increased the availability of and access to information. Although computers have rich output capabilities such as screens, sound and haptics, they lack input richness. Most systems are driven by a limited set of explicit input modalities (e.g., typing, touch, voice). Today we see a proliferation of new computer devices with a diversity of sizes and proximity. As these devices shrink is size, they become more portable and can be used to access information while on-the-go; However, these limited explicit input modalities are less useful in these scenarios. Our personal computing devices demand interaction techniques that treat these differences in size not as constraints but opportunities to devise novel interaction techniques for them.

Virtual and augmented reality (VR and AR) have shown great promise to address this problem. Paired with a rapidly growing user base, it will likely be the next ubiquitous device (after smartphones) that fundamentally changes human-computer interaction. To support the growing interest towards AR/VR technologies, there has been a significant amount of research on interaction techniques that allow users to easily and naturally manipulate content in the AR/VR space.

Today's AR/VR systems rely on either handheld controllers, which have been the primary tool for interactions with artificial worlds, or in-air bare hand gestures. In-air hand tracking excels at coarse-grained gestural input but supports limited fine-grained interactions. On the other hand, although handheld controllers are useful input devices and they offer advantages such as physical buttons with tactile feedback [18, 140] and most importantly, a mechanism for precise tracking of the hand, they do not support mobility. If head-mounted displays are to become a compelling personal computing platform, they need to be always carried (or worn) by the user, so that we can use it anytime while performing other activities. In this thesis, I focus on non-vision based methods to enable new input devices that power more robust handheld controllers and wearable devices that power subtle finger and wrist-based interactions.

1.2 Research Overview

Conventional input techniques rely on hand-held controllers or in-air gestures. These devices are designed and optimized to track the hand and arm in front of the body [87, 8, 135, 127, 79, 101]. Such systems make compelling demos, but prolonged use of such large arm movements lead to fatigue. Smaller movements improve efficiency, reduce fatigue, and broaden possible usage scenarios. However, they should be accompanied by precise tracking for high-fidelity interaction and must be continuously available and subtle to support various contexts of use while enabling robust and expressive interaction. While there are numerous ways to provide input to a wearable computing device, a few classes of input stand out as promising candidates for the small, yet precise input needed on head-mounted displays.

As discussed, fully tracked **handheld objects** can be used as a reliable input device for AR/VR systems. Existing optical tracking solutions require the use of bulky controllers with optical sensors that cannot be occluded by the body. Commonly used outside-in tracking systems require external infrastructure which limits their use to a particular space [99, 21, 135, 127, 79, 101]. However, for mobile usage, these devices must use inside-out tracking and be usable with the hands at the sides and out of direct line of sight of a head-mounted camera. They must also be small enough to fit in a pocket, which precludes the use of bulky tracking rings.

The expressivity of human **wrist** is an innate ability of ours, but an under-explored method of interaction in AR/VR settings. A user can provide continuous input using their wrist, in eyes-away contexts where line-of-sight to a camera might be difficult to maintain.

Wearable devices that sense small **finger** motions can enable both direct and indirect manipulation techniques. Such devices are particularly well-suited for augmented reality computing platforms. When tracked in absolute coordinates, designers can re-purpose existing objects and surfaces in the environment as interactive elements. For example, a users can provide text entry through hand drawing using their dominant hand on the back of the non dominant hand. Such interactions are less susceptible to fatigue because the user is touching an actual physical surface that supports their arm, instead of reaching out and touching in mid-air.

1.3 Challenges and Problems

Although the emerging market of various computing devices provides a tremendous opportunity to introduce novel, efficient, and user-friendly input modality, there are still different challenges from a device perspective and user perspective, which need to be overcome.

1.3.1 Design for low power consumption

Tracking the movements of the hands, fingers, or in general any object can be done by multiple modalities including camera-based [57, 78], ultrasound [51] or impedance tomography [159]. Each of these methods have their own unique advantages; however, these sensing techniques typically require high energy consumption. In this thesis, I have explored sensing modalities and have designed hardware solutions that particularly result in a tracking system that have low power consumption and can be operated by a small battery.

1.3.2 Design for continuous tracking - beyond gestural tracking

Many of today's most effective input devices rely on continuous tracking of the fingers or handheld objects; smartphones and smartwatches track 2D finger position, a mouse tracks 2D hand position, and augmented reality devices like the Microsoft HoloLens 2 track the 3D pose of the finger. These proven input techniques stand in contrast to existing lines of research around sensor-based efforts to track finger or wrist motion, which often focus on discrete, gesture-based interactions[36, 159, 37, 102]. Although gesture recognition is useful, it is just one part of the interaction language needed for mobile wearable computing. For common tasks like object manipulation, drawing, sliding, swipe-based text input, or body-pose reconstruction, an input device that supports absolute, continuous tracking with millimeter-level precision will be most appropriate. Moreover, building a gesture recognizer on top of a tracking system, instead of direct classification from a sensor stream, can enable input devices that are more robust and extensible (e.g. supporting different users and contexts out of the box).

1.3.3 Minimal calibration

Most bio sensing systems[51, 159, 35], need per-session calibration to perform well. However, an ideal input device should work out of the box, without any user calibration. Therefore, in this thesis, I have prioritized robustness and accuracy and designed the system from ground up to support continuous, absolute tracking with minimal calibration. For some of my work as I discuss in chapter four and six, the use of a physics and sensor model is a significant advantage of tracking algorithm, compared to other methods. It allows simulation and experimentation to optimize system configuration and decouples model training from user evaluation. That said, our finger tracking approach does still require the use of a one-time factory calibration. For the prototype presented in chapter four, such a calibration is essential, due to the manufacturing tolerances of the device assembly. However, it is possible that for a mass-produced device, the tolerances can be made tight enough such that a single model would suffice for all units.

In chapter five, by using a simple and regression model compared to deep neural networks, we were able to deign a system that not only can work across users, but with a very small per user calibration, the tracking accuracy gets significant better.

1.4 Thesis Organization

In this thesis, I argue that continuously available input devices that require minimal calibration can unlock new capabilities and enable applications that are more natural and subtle through the use of new hardware designs. In support of this thesis, the subsequent chapters are presented:

1. *RQ1: How can we bring 6-DoF tracking to small handheld objects?*
   In chapter three, I present Aura [142], an inside-out electromagnetic tracking technique for small, handheld VR controllers.

2. *RQ2: Can we enable wrist tracking using a A simple, low-power hardware architecture? Can this be used as a pointing device and how would it compare to a high-precision wrist and forearm tracker?*
   Chapter five presents RotoWrist, an infrared (IR) light-based device that performs continuous 2-DoF wrist tracking from the wristband.

3. *RQ3: What interaction techniques and hardware can be used to support productive use cases using subtle, finger-based input? How would such new input events be used for wearable interaction?*
   Chapter four presents AuraRing [90], a wearable magnetic tracking system designed for tracking fine-grained finger movement. In chapter six, I present BiRing, a hand to hand 2-DoF relative tracking system using two ring-worn devices.

Chapter 2

RELATED WORK

Since several of the projects in this thesis rely on electromagnetic tracking, I will first elaborate on advances in electromagnetic tracking techniques (Section 2.1). The rest of this chapter contains a broad overview of research efforts in tracking techniques for handheld controllers (Section 2.2), methods for finger (Section 2.3.1), wrist angle (Section 2.3.2) and hand to hand relative pose tracking (Section 2.3.3).

2.1 Magnetic Tracking

The use of electromagnetic (EM) fields for 6-DoF tracking has a rich history that dates back to the 1980's [62, 98]. This technique is characterized by extremely precise position and orientation tracking. Since then, it has been used in contexts ranging from surgery [31, 63] to biomechanics [77, 92] to localization [17].

In general, magnetic tracking systems rely on two types of sources: the permanent magnet [76], which generates a static magnetic field, and alternating current (AC) electromagnetic coils [118, 117]. While permanent magnet systems are affected by the Earth's geomagnetic field, AC electromagnetic coils generate a magnetic field at a particular frequency that can be more easily measured by filtering out all other frequencies.

A typical electromagnetic tracking system contains a transmitter base station with a three-axis orthogonal coil transmitter and one or more three-axis receivers, typically realized using orthogonal coils. The oscillating magnetic flux through the coils induce an oscillating voltage of the same frequency which can be amplified and measured. Filtering or synchronous detection is used to isolate the field of interest from other ambient magnetic fields, including the geomagnetic field.

Using an alternating current to drive an electromagnet formed by a wire coil is an effective method of generating a magnetic field that oscillates at a particular frequency. According to Maxwell's equations, an electric current flowing along a wire coil will generate a magnetic field. The oscillating magnetic flux from these generator coils intersects the sensor coils, inducing a voltage of the same frequency according to Faraday's law of induction. The voltage induced in the coil is proportional to the rate of change of the magnetic flux through the surface bounded by the coil and the number of windings in the coils.

The flux through the coil depends on its orientation within the magnetic field. If the coil is aligned with the field—that is, the normal vector to the coil is aligned with the field—then the flux and the magnitude of the induced voltage will be maximal. As the coil rotates away from the field, the induced voltage decreases to zero. If the coil flips around, the voltage acquires a 180° phase shift, which could manifest as a negative amplitude, depending on the measurement technique.

To track position and orientation in real time, researchers have historically relied on tracking pose changes [98] or on iterative algorithms that find the pose which best explains the observed sensor values [95] These systems are usually realized by approximating the magnetic sources as dipoles and, in multi-axis systems, as orthogonal dipoles. These pose estimation approaches rely on analytical or numerical analysis of forward models, which describe the magnetic field at the sensor as a function of pose. More recently, closed-form solutions [59] have been developed that analytically invert the forward models. In all of these approaches, deviations from these ideal models cause inaccuracies in the estimated pose.

These assumptions lead to traditional electromagnetic tracking systems that use large high-power transmitter coils that can be approximated as dipoles and are intended to be placed in the environment away from any metallic elements. In contrast, in this thesis I try to forgo these constraints and designed input devices that are optimize for form-factor, power and precision.

As I mentioned earlier, magnetic tracking can be classified as DC and AC tracking. In DC tracking systems, a magnetic sensor, often a magnetometer, measures the magnetic field generated by a permanent magnet. This technique has seen extensive use in the HCI field, since magnetometers are commonly found in wearable devices and permanent magnets can be easily attached to the body.

Dai et al. demonstrate an electromagnetic tracking technique using a single transmitter coil and a 3-axis magnetometer [24]. In Finexus, Chen et al. use four magnetometers to track the position of electromagnets placed on the fingertips [16]. Both of these approaches are limited to shorter distances (less than $20\,\mathrm{cm}$). Islam et al. show a technique using resonance coupling to improve the efficiency of electromagnetic tracking systems [52]. Our experimentation with this technique suggests that while range is increased, the crosstalk due to the mutual coupling of different coil axes makes field reconstruction difficult.

Abracadabra [44] used a permanent magnet affixed to the fingertip to interact with a smartwatch that measures the magnetic field direction and strength using a magnetometer. Similarly, uTrack [15], affixed a permanent magnet to the thumb and two magnetometers to the fingers to track 3D thumb pose. FingerPad [14] also affixes a magnet to thumb but uses Hall-effect sensor grid on the index fingernail to sense pinch gestures. The Nenya system [7] used a ring with two permanent magnets attached that could be twisted to change the shape of the resulting magnetic fields, which were measured by a magnetometer on a smartwatch. In 2016, Lyons showed a smartphone-based magnetometer tracking system that enabled 2D interaction on the side of a cardboard VR device using a permanent magnet [71]. Reyes et al demonstrated SynchroWatch [102], which recognizes synchronous thumb gestures using a permanent magnet attached to a thumb ring.

Magnetic tracking systems that rely on permanent magnets have size and complexity advantages because they can rely on off-the-shelf magnets and magnetometers. However, this comes at the cost of precision. Magnetometers are generally limited in their accuracy and sampling rate and such systems must contend with interference from the Earth's geomagnetic field. As a result, the operating range of such systems is generally limited to several centimeters.

In contrast, AC magnetic tracking relies on one or more oscillating magnetic fields. AC electromagnetic tracking has a rich history of enabling precise, 6-DoF tracking [98, 62]. Since its inception in the late 1970's, the technique has been used for surgical tracking [63, 31, 147], biomechanical analysis [92, 77], virtual reality tracking [59, 143], and HCI [93, 16].

Original systems relied on iterative approaches to estimate the pose of the tracked objects, however, recent techniques have explored closed-form or closed-loop solutions that improve performance [59, 32]. Ge et al. use rotating transmitters [32] to continuously track the object. Kim et al. show a closed form solution when using a 3-axis dipole generator [59], but only evaluate their approach on a small 3D trajectory with no ground truth reference. These approaches rely on analytic solutions to the forward and reverse problems.

Magnetic field distortions when tracking near metallic objects is a common drawback of AC magnetic tracking, particularly when used in dynamic environments. Some work has attempted to account for magnetic interference through a secondary calibration step [60, 61].

Unlike most EM systems, which rely on environmentally placed transmitter coils, Pirkl et al. developed a wearable low-power electromagnetic system [93]. However, this solution was only used for gesture recognition, not positional tracking. Roetenberg et al. created a wearable EM tracking system with $5\,\mathrm{mm}$ accuracy [103] using a pyramidal structure of transmitter coils. For a more thorough review of magnetic positioning systems, see [91].

In the commercial space, electromagnetic tracking is performed by products from Polhemus, NDI, and Sixense. These products consist of a large transmitter that emits a consistent magnetic field that spans a volume on the order of a cubic meter. The base stations are coupled with small receiver sensors that plug into a large processing hub. In general, such devices offer incredible tracking precision and accuracy, but rely on large infrastructure and distortion-free fields that make integration into a mobile device difficult. For example, the Polhemus G4 [96] tracking system uses a $5\,\mathrm{W}$ transmitter and $2.5\,\mathrm{W}$ receiver hub.

Magic Leap has recently released a proprietary electromagnetic tracking solution. However, this system uses a multi-frequency transmitter in the controller while placing the receiver coil in the headset. This configuration makes it difficult to scale to multiple controllers, as nearby controllers may interfere with each other.

2.2 Inside-out Controller Tracking

Some commercial devices (Windows Mixed Reality, Oculus Quest) use head-mounted cameras that track a controller with an external LED ring. While this technique is precise, it requires the use of bulky controllers and line of sight to the headset, preventing use with arms at the side. The upcoming Vive Focus uses ultrasound and IMU tracking, though there are few details about how this works.

In the research space, a number of different techniques have been explored to eliminate the use of bulky tracking rings and markers. Pocket6 is a solution that uses ARKit on an iPhone X to localize the controller using the built-in SLAM and IMU fusion algorithms [8]. Pandey et al. demonstrate a technique to track markerless controllers using only the front-facing camera on the HMD [87]. These classes of solution tend to demand significant power and computation.

Though not explicitly a controller tracking technique, Shen et al. show a method [111] to track the position of a smartwatch using IMU sensors and a kinematic model. Ultrasound tracking systems have long been used for tracking the head [25, 30]. These rely on pairs of beacons and microphones that use time-of-flight measurements to estimate 6-DoF pose. Though they are usually small and light, they are sensitive to temperature, occlusion, and ultrasound noise. Nandakumar et al. use RF backscatter to track the position of sub-centimeter scale devices [82]. Although they can localize objects tens of meters away, their accuracy is not sufficient for VR controller trackers and requires instrumenting the environment. See Baillot et al. for a more complete review of VR tracking technologies [9].

2.3 Wearable Devices

2.3.1 Finger Tracking

Magnetic Tracking Approaches

In addition to research advances, there are several commercial AC magnetic tracking solutions on the market. Systems from Polhemus [96] and NDI [85] offer sub-millimeter tracking accuracy at a range of dozens of centimeters. Both the Razer Hydra and the Magic Leap One use electromagnetic tracking for handheld controllers

Smart Rings

Smart rings provide an attractive form-factor for always-available subtle input. Rings have been shown useful for subtle input [7], interaction with 2D surfaces [151, 80, 55]. LightRing [55] uses a gyroscope and IR proximity sensor to track finger motion on a 2D surface. Commercial smart rings offer features like fitness tracking, heart-rate tracking, inertial gesture sensing, and NFC payments. CyclopsRing, which uses a fisheye lens camera clipped between the fingers to detect one handed gestures or touch gestures from the user's other hand [13]. PeriSense, detects gestures of the finger wearing ring and its adjacent fingers by measuring capacitive proximity between electrodes and human skin [144]. QwertyRing presented a technique that supports text entry on physical surfaces using an IMU ring [38]. ElectroRing is a ring-based input device that reliably detects both onset and release of a subtle finger pinch, and more generally, contact of the fingertip with the user's skin [56]. TelemetRing proposed an inductive telemetry approach eliminating bulky batteries or capacitors from the ring part that supports command and text entry in daily lives by detecting finger typing on various surfaces [126].

Other Approaches for Finger Tracking

Beyond magnetic tracking approaches, researchers have leveraged other sensing techniques to estimate fine-grained finger motion. Kim et al used infrared illumination from the wrist to track hand pose [58]. Project Soli [137] uses radio frequency signals to recognize subtle finger gestures. Pyro uses the pyroelectric effect to recognize small thumb gestures [37]. Bioimpedance has been used to reconstruct the cross-sectional impedance of the arm and recognize various hand gestures [159, 160].

Other systems use sensors on the opposing arm or in the environment to track finger motion. FingerIO uses sonar from a smartphone to measure the 2D position of the fingertip [83]. SkinTrack uses a ring-smartwatch pair that leverages electrical waveguides to track the 2D position of the finger on the opposing arm [161].

Still other systems use gloves or finger strips to estimate finger and hand pose [155, 154, 47, 49, 141, 5, 33, 2, 125]. These systems require significant augmentation and are less appropriate for everyday usage.

Finally, camera-based approaches are a common approach to hand- and finger- tracking. In this space we have These commonly rely on infrared depth sensing [127] or markers on the hand [136]. Unique placements of the cameras have lead interactive systems that offer subtle input [13, 57, 69]. TouchCam combines data from infrared sensors, inertial measurement units, and a small camera to classify body locations and gestures using supervised learning [121].

2.3.2 Wrist Angle tracking

Existing hand tracking methods are based on sensors that are either outside-in (off-body) or inside-out (on-body). Outside-in sensors with cameras [99, 21, 135, 127, 79, 101] or radio frequency[134, 68] employ external sensing infrastructure that limits users' mobility. In comparison, inside-out sensing approaches provide much better mobility support. Therefore, in this section we have focused on inside-out wrist tracking and have split the related work into two main categories: 1) tracking wrist movement by leveraging the internal anatomy of the wrist and 2) tracking using the external contours of the hand.

Inferring wrist angles from internal signals

More often, existing lines of research around wristband sensor-based efforts to track hand motion focus on discrete, gesture-based interactions. Among those, a number of research projects explored electromyography (EMG) where electrodes are placed below the elbow to detect hand gestures [54, 107]. EMG systems are complex and require the user to wear a bulky array of electrodes below the elbow. Other researchers have leveraged electrical impedance tomography [159], bio-capacitive sensing [100, 129], IR tomography [73], acoustic tomography [45, 27] and more recently ultrasound [51] for discrete hand gesture recognition from a wristband. These methods result in discrete, gesture-based interactions like detecting a fist vs stretch, directional sweep, finger pinches etc. and not continuous tracking which is an important component of tasks like target selection. Furthermore, these systems are often very sensitive to the positioning and potential slippage of the device and therefore often require per-session training.

Inferring wrist angles from external signals

Skin surface deformation due to wrist movement can be detected by pressure/stretch sensors [53, 26, 124]. Other researchers have leveraged ultrasound beam forming [50] and acoustics [83, 64] to recognize discrete gestures. Other approaches to determining hand pose on wearable devices employ vision sensors such as a camera or optical sensor on the inner side of the arm or wrist [57, 153, 36, 145]. Digits [57] used infrared illumination from the wrist to track hand pose. Opisthenar uses an embedded wrist camera to recognize static hand poses [153]. Although this method can enable full hand tracking, the form factor and power consumption limits practicality. FingerTrak use 4 thermal cameras around the wrist to reconstruct full hand pose though can not reconstruct relative angle of the wrist with respect to the forearm [48].

2.3.3 Hand to hand relative pose tracking

Gestures involving two hands include more degrees of freedom and are more efficient [11] than one-handed gestures in many interaction tasks. Gestures that require both hands to cooperate with no contact are often customized for specific input vocabularies, such as graphical user interfaces (e.g., pan and zoom gestures [81, 110, 131], object manipulation (e.g., rotating and scaling in AR [94]) and mid-air drawing [41]. These gestures are usually axisymmetric or centrosymmetric, following humans' natural motor behavior and have been assigned conventional semantics in daily life (e.g., zoom in/out).

In order to utilize the mutual control and bidirectional haptic feedback between two hands, many previous works have studied bimanual gestures which allow two hands to come into contact with one another. One idea is to treat the region on/around the non-dominant hand (e.g., palm [42, 10, 133], back of the hand [120, 157, 158, 161, 162], forearm [109, 148]) as a touchable surface and allow the dominant hand to perform on-skin gestures (e.g., tap, draw, pinch, etc.). Some researches have also studied touch accuracy and learnability on different locations of the hand [43] and pointed out that landmarks on the hand (e.g., fingertips, metacarpal bones) are easy to touch and remember[138]. Lu et. al. also explored the design space of hand-to-hand gestures by designing a group of gestures that are performed by touching one hand with the other hand using two off-the-shelf wrist-worn devices [70].

Chapter 3

INSIDE-OUT 6-DOF ELECTROMAGNETIC CONTROLLER TRACKING

3.1 Introduction

Head-mounted displays (HMDs) for virtual (VR) and augmented (AR) reality represent a promising direction for next-generation computing platforms. Although early iterations of such devices are intended for use in fixed environments, the promise of mobility offers exciting new opportunities for the future of personal computing. The handheld controller is a useful input device and offers advantages such as physical buttons with tactile feedback, a platform for rich haptic feedback [140, 156, 123, 19], and, most importantly, a mechanism for precise tracking. Controller tracking is classified as either 3-DoF rotational tracking (i.e. roll, pitch, yaw) or 6-DoF rotational and positional tracking (i.e. roll, pitch, yaw, x, y, z). The use of 6-DoF positional tracking enables an additional class of spatial computing applications by allowing the controller to serve as a virtual tool or a proxy for the user's hand.

Today, there is a significant divide between handheld controllers for mobile VR/AR platforms and those for high-end desktop VR systems. Mobile systems, such as the Samsung Gear VR or Google Daydream, use a small handheld controller that relies on 3-DoF inertial orientation tracking, with limited positional tracking support. On the other hand, desktop VR systems like the Oculus Rift, use a larger controller with external light emitting diodes (LEDs) for 6-DoF optical tracking. These external elements are tracked using cameras placed in the environment. Though these controllers are much larger, the positional tracking they offer results in a substantially more immersive experience compared to the simple orientation tracking on mobile VR. Efforts to remove the need for environmental infrastructure often move the tracking cameras to the head. This approach leads to additional HMD power consumption, limits cameras' line of sight, and still requires bulky tracking elements on the controller.

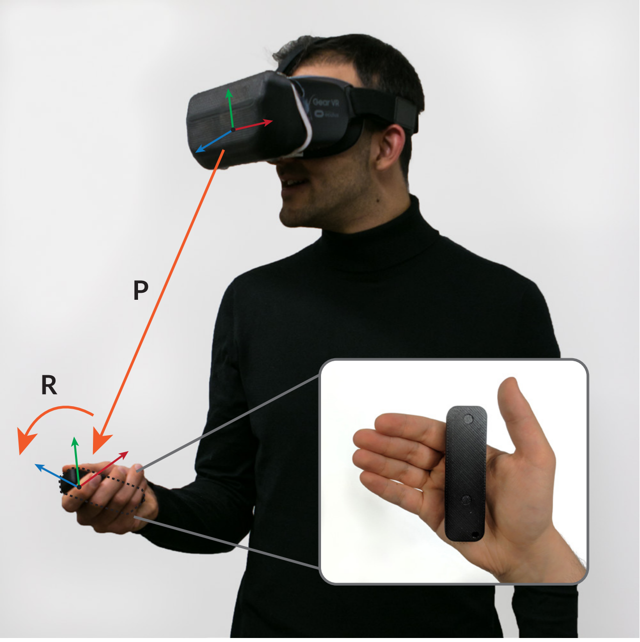


Figure 3.1: Aura is a 6-DoF electromagnetic tracking system for small, handheld controllers. It estimates the pose of the controller with respect to the head-mounted display.

If head-mounted displays are to become a compelling personal computing platform with applications beyond gaming, they must support mobility and robust usage. Handheld controllers must rely on inside-out tracking to enable mobility while maintaining low power consumption for extended use on battery power. The form factor should be small enough to fit within the hand during use and in a pocket or bag when not in use. They should also support a robust set of interaction scenarios and maintain usability outside of the view of a head-mounted camera—for example, with the hands at the side on a crowded bus or under a table during a meeting. In this chapter, I seek to close the gap between the attractive form-factor and mobility of 3-DoF inertial controllers and the performance and usability of high-end 6-DoF controllers.

We propose Aura, a novel low-power electromagnetic tracking technique to bring high-precision 6-DoF controllers to any head-mounted mixed-reality system without the need for line-of-sight, bulky tracking rings, or environmental sensors. Our proposed tracking system uses three coils embedded in a head-mounted display that each generate a unique magnetic field oscillating at $100\,\mathrm{kHz}$. The generated fields induce a sinusoidal voltage in orthogonal receiver coils embedded in a handheld controller. As the user moves the controller, the signal in each receiver coil varies depending on its position and orientation within the field.

In a traditional electromagnetic tracking system, one would carefully construct a 3-axis dipole transmitter and place it far away from any other metallic objects. This significantly limits the design space of HMDs, which strive to be small and lightweight. Unlike traditional electromagnetic tracking systems, Aura makes no assumptions about the size, shape, or position of the transmit coils and no assumptions about nearby ferromagnetic material on the headset. This freedom enables custom coil shapes and configurations that open the design space for small and lightweight form factors. To demonstrate this capability, Aura foregoes the use of precisely manufactured orthogonal transmit coils and explicitly uses irregularly shaped coils designed to fit the contours of a Samsung Gear VR headset. The use of irregularly-shaped coils significantly complicates the pose estimation task. To solve this challenge, Aura uses a hybrid tracking approach that leverages a physics model to calibrate the sensor coils and a closed-form data-driven model using neural networks to directly estimate pose from the calibrated sensor data. While electromagnetic tracking has a rich history [98, 62, 31, 63, 77, 92, 17], to our knowledge, Aura is the first system to demonstrate 6-DoF pose estimation using irregularly-shaped, non-orthogonal coils.

Aura is inside-out; that is, it tracks the position of the controller with respect to the head. For high-end AR or VR systems with positional head-tracking, e.g. using inside-out SLAM-based tracking [46], Aura provides an upgraded controller tracking experience by enabling more robust and subtle usage without the need for line-of-sight or bulky tracking rings. On low-end VR systems that rely on inertial head-tracking, the ability to locate the hand with respect to the eyes still enables many new interactive experiences. On such devices, Aura serves as a snap-on upgrade that provides positional tracking of the controller. Positional head-to-controller tracking allows a VR system to render the hand or virtual objects within the hand at the correct visual position and orientation no matter how the user moves their head. This capability would allow the user to perform any of the pointing-based interactions common in today's VR applications, without having to frequently recalibrate to compensate for drift. Moreover, it would allow the user to directly manipulate objects locked or loosely locked to the body. For example, a user might reach out and type on a virtual keyboard placed directly in front of them, a task that is nearly impossible with inertial tracking. We note that on low-end systems with only inertial head-tracking, the ability to directly grasp other objects located in the world is still limited by the system's head-tracking capabilities.

In the following sections, I provide a brief overview of electromagnetic tracking techniques and design decisions I made in Aura, implementation and calibration details for the Aura system, and results from a system evaluation and characterization. Our results demonstrate that Aura can track a handheld controller with millimeter accuracy.

Specifically, our contributions include:

1. An efficient, novel, closed-form tracking algorithm that works with arbitrary transmitter coil shapes and configurations and accounts for static magnetic field distortions
2. A low-power hardware architecture for a 6-DoF handheld controller
3. A prototype implementation of the Aura system and evaluation of tracking accuracy that demonstrates a median 3D error of $5.5\,\mathrm{mm}$ and $0.8°$.

3.2 System Implementation

The Aura system consists of an HMD attachment that generates magnetic fields and a controller that measures the resulting fields. Figure 3.2 depicts the Aura system components. The following sections provide details of the Aura hardware and explore the capabilities and design challenges of the devices.

3.2.1 Transmitter

Aura's transmitter consists of three low-profile generator coils embedded in a head-mounted display that sequentially emit a magnetic field oscillating at $100\,\mathrm{kHz}$. Each of the side generator coils consists of 30-40 turns of 22 AWG magnet wire wound around a 3D printed ABS

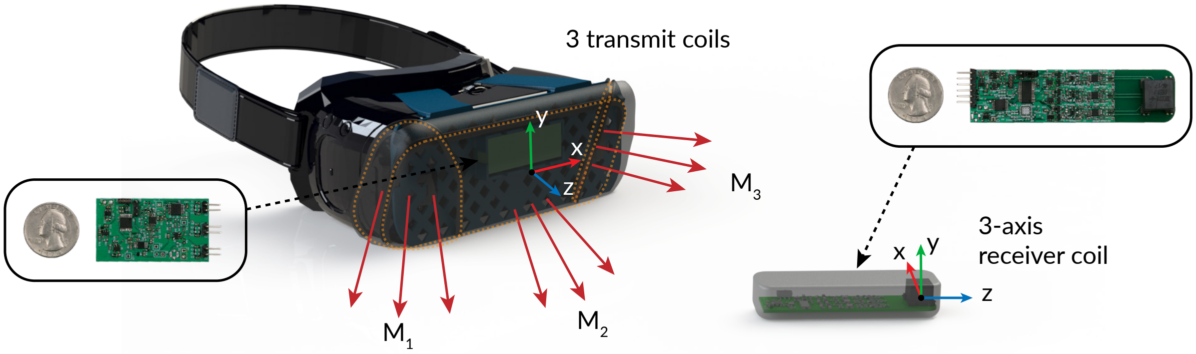


Figure 3.2: Three transmitter coils embedded in the headset produce three magnetic fields (M_1, M_2, M_3). A three-axis coil inside the handheld controller measures the fields.

frame. The central coil is wrapped in an oval shape of size $20\,\mathrm{cm} \times 7.5\,\mathrm{cm}$. The inductances for the three coils are measured as $358\,\mu\mathrm{H}$, $476\,\mu\mathrm{H}$ and $279\,\mu\mathrm{H}$.

One of the contributions of this chapter is the use of non-orthogonal coils for the transmitter. This allows Aura to have a configuration of coils that can be fit into any HMD. The

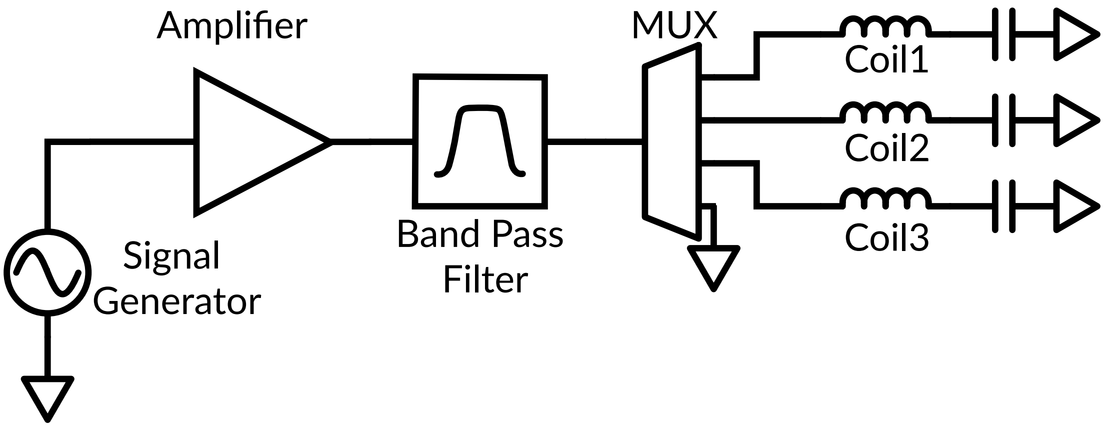


Figure 3.3: The transmitter filters the output of a waveform generator and feeds the signal through each of the transmitter coils sequentially.

coils are rigidly mounted on a 3D-printed support structure such that the magnetic flux is directed toward the user's hand. Our implementation is designed to fit on a Samsung Gear VR (Figure 3.2) but could easily be modified to fit other HMD designs.

Figure 3.3 shows the block diagram of Aura's transmitter. The transmitter uses a programmable waveform generator (AD9833) to generate a $100\,\mathrm{kHz}$ square wave. This frequency was chosen to enhance sensitivity without approaching any of the coils' self-resonant frequencies or inducing troublesome eddy currents in nearby metallic objects. This signal is passed through a two-stage active bandpass filter (AD8616) with a Q-factor of 15.9 and gain of $0.5\,\mathrm{dB}$. The resulting sinusoid is time-multiplexed (ADG1604) and fed through each of the generator coils.

Each coil generates a magnetic field for $3\,\mathrm{ms}$ in sequence; once the cycle is completed, all of the channels are turned off for $2\,\mathrm{ms}$ to synchronize the transmitter and receiver (Figure 3.4). Because Aura cycles through each coil every $11\,\mathrm{ms}$, the resulting frame rate is $91\,\mathrm{Hz}$. An ultra-low power microprocessor (MSP430FR2100) controls the multiplexing and interfaces with the components on the transmitter board. We use passive matching networks to tune the coils impedance, resulting in improved power transmission efficiency at $100\,\mathrm{kHz}$.

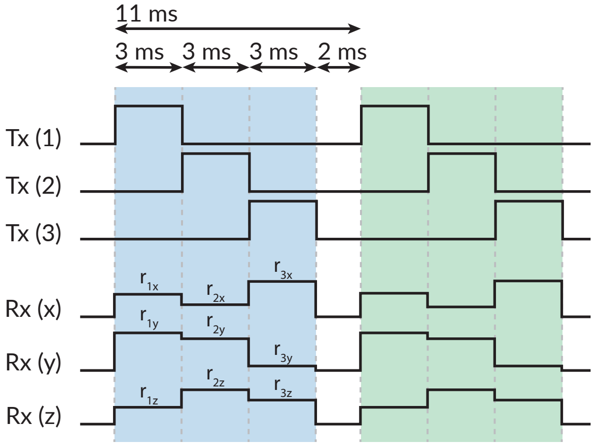


Figure 3.4: Each transmitter coil is activated for 3 ms, during which each of the three orthogonal receiver coils measures the magnetic flux.

3.2.2 Receiver

While the HMD-based transmitter uses hand-wound coils to generate three unique AC magnetic fields, the controller uses an off-the-shelf three-axis orthogonal receiver coil (Grupo

Premo 3DCC08) to reconstruct the 3D magnetic field vector. Because the tracking system relies on interpreting the demodulated sensor measurements as field vectors, it is important to maximize precision and orthogonality in this procedure. The signal from each of the coils is fed to an amplifier (INA826) with a gain of $33.9\,\mathrm{dB}$. The resulting amplified signal is fed to a two-stage active bandpass filter (AD8616) with a Q-factor of 10.2 and a gain of $33.5\,\mathrm{dB}$. Then, I use a low-noise and low-voltage drop Schottky diode network (SMS7630) in a full-wave bridge rectifier configuration[39] to demodulate each of the channels. Finally, each channel is sampled at $4\,\mathrm{kHz}$ using the 24-bit sigma-delta ADCs of an MSP430i2031. Figure 3.5 (Top) summarizes the analog signal processing pipeline in Aura's receiver.

This method effectively captures the magnitude of the magnetic fields but does not resolve the phase of the oscillating fields. Reconstructing position or rotation would be very difficult from this approach since rotating the controller $180°$ along one of its axes would result in the same overall magnitude for each of the coils but with a $180°$ phase shift on two of the axes. Since Aura uses a three-axis receiver and each axis could be in- or out-of phase with the transmitter, there are $2^3 = 8$ possible vectors that would deliver the same sensor values. For each frame of three vector fields (one from each transmit coil), there are a total of $8^3 = 512$ possible rotation states given the same channel magnitudes.

Aura uses a low-power solution to reduce this sign ambiguity. First, comparators binarize each of the amplified signals from the receiver coils prior to rectification. The comparator outputs logic low when the amplified signal of the channel is less than its common-mode voltage ($V_{CM} = 1.2\,\mathrm{V}$) and logic high otherwise. An XOR gate estimates the relative phase between a receiver channel and a reference signal. The output of the XOR gate is low when the two signals are in-phase and high when the two are out-of-phase. One can produce a referenced signal locked to the transmitter using a phase locked loop, but to save power and simplify the design, I have chosen channel 1 of the receiver as the reference. The resulting logic signal is low-pass filtered by an RC network to remove any glitches due to phase mismatch. These digital signals are then sampled by GPIO pins of the microcontroller unit (MCU). Using this low-power approach, the ambiguity of the signs reduces to $2^3 = 8$

possible solutions since the phase of the reference channel with respect to the transmitter is still unknown. The exact signs can be determined by placing the controller in a known start state and temporally filtering to ensure consistency over time. Figure 3.5 (bottom) summarizes Aura's sign detection capabilities.

With the amplitude demodulation and phase estimation components, the MCU now has access to a signed magnitude for each channel every $250\,\mu\mathrm{s}$ ($4\,\mathrm{kHz}$). This data stream contains the time multiplexed transmit signals and resembles the diagram in Figure 3.4. A segmentation step recovers each of the 9 measurements from this data stream. The seg-

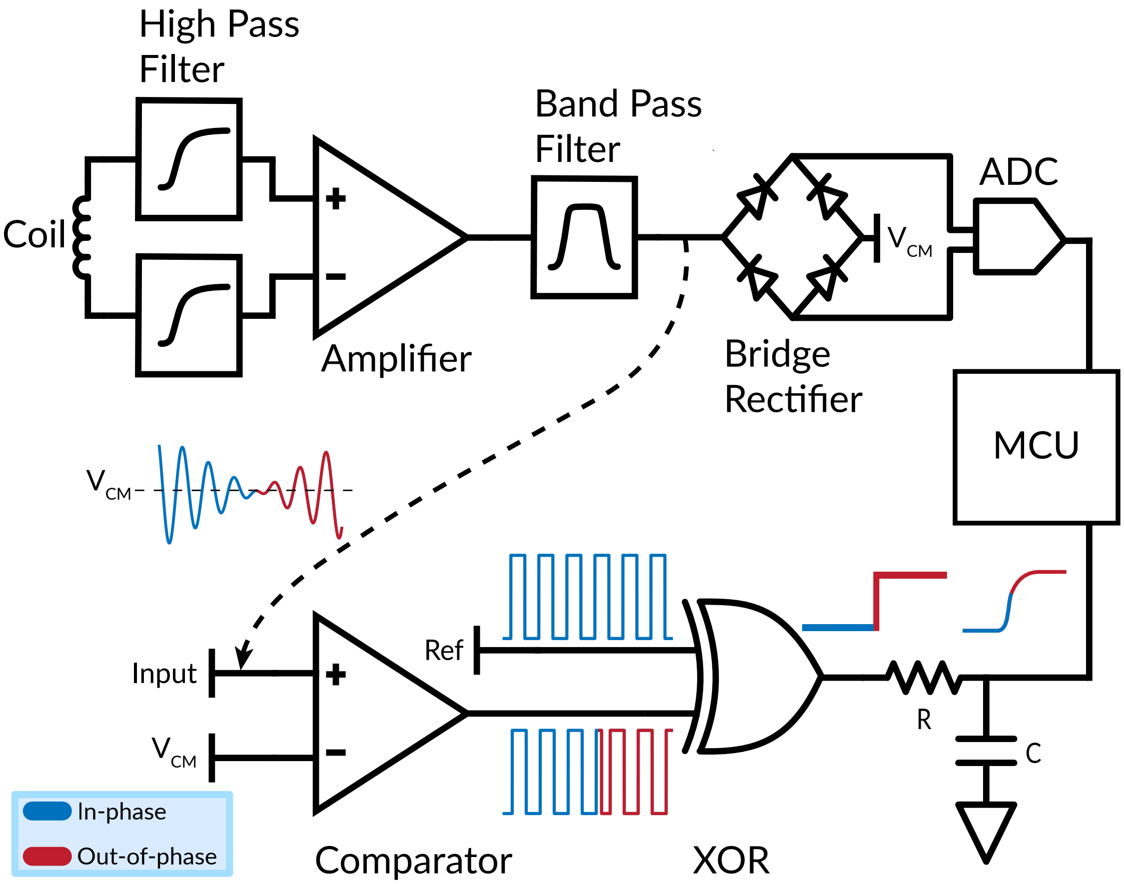


Figure 3.5: On the controller (single-channel depicted), the MCU measures the magnitude (top) and phase (bottom) of the received signal.

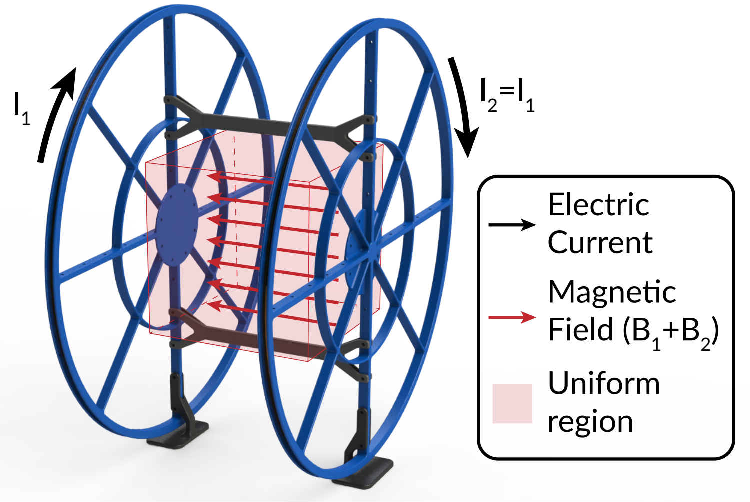


Figure 3.6: For calibration, Helmholtz coils produce a small volume where the magnetic field is insensitive to movement.

mentation algorithm will search for a local minimum to find the “off state” and synchronize itself with the transmitter. It then uses the known timing of the signal to extract the 9 mean values of each coil measurement ($r_{1\{xyz\}}$, $r_{2\{xyz\}}$, and $r_{3\{xyz\}}$). The MCU then sends the nine reconstructed signals to a PC over USB. In software, a digital second-order Butterworth filter with a $10\,\mathrm{Hz}$ cutoff frequency is applied to the raw signals for further noise reduction.

3.3 Calibration

In order to treat the measurements from the ADC as magnetic field strengths, a calibration procedure must be performed to account for imperfections in the signal processing chain and channel gains. This calibration is intended to be performed only once per device, i.e. through a factory calibration step. While calibration in magnetic tracking systems often refers to modeling magnetic field distortions, Aura inherently accounts for this in the tracking algorithm described in Section 3.4.

To assist in the calibration process, a set of Helmholtz coils as shown in Figure 3.6 is constructed. With this device, current through two parallel coils (I_1 and I_2) generate AC magnetic fields. Because of the spacing of the coils, the tangential components of the fields cancel, and the coil pairs create a small volume with a uniform magnetic field, indicated by the red shaded region in Figure 3.6. Within that region, the magnetic field direction and magnitude are relatively insensitive to the receiver coil's position. By controlling the amplitude of the AC current through the coils with a function generator, we can precisely control the magnetic field strength within this volume.

3.3.1 Signal chain modeling

In an ideal sensor, the measurement for any axis would be linearly correlated with strength of the magnetic field along that axis. To measure this linearity, I place the Aura controller within the Helmholtz coil and use the function generator to step up the magnetic field strength linearly in small increments and record the Aura measurements. Figure 3.7 shows the observed signal and the ideal linear response. Though the signal is linear for much of the observed field strengths, significant nonlinearities were observed in the presence of weak magnetic fields.

We model these observations using three parameters: a gain term (g), Gaussian noise (n) inherent to the signal chain, and a bias term (b) due to the forward voltage drop across the diodes. Equation 3.1 summarizes how these effects influence magnetic field strength (f) to produce the Aura measurement (r).

$$r = g\sqrt{f^2 + n^2} - b \tag{3.1}$$

After collecting data from the Helmholtz coil, I use an optimization procedure (SciPy) to fit this model to the observed data. As shown in Figure 3.7, this model is a good fit for the observed data. We then invert this model to derive an expression, as shown in Equation 3.2, for the desired magnetic field strength (f) as a function of the ADC measurements (r). We

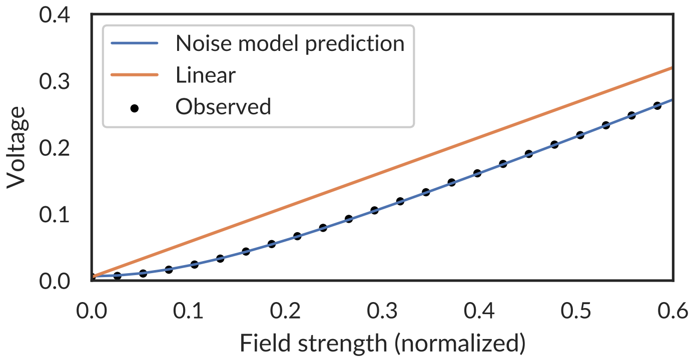


Figure 3.7: Our proposed model outperforms a linear prediction of the observed signals.

preserve the sign of the original signal as described in Section 3.2.

$$f = \sqrt{\left(\frac{r+b}{g}\right)^2 - n^2} \tag{3.2}$$

At runtime, I further improve the device performance by assuming the bias and gain terms remain constant and dynamically adjusting the noise term of this model based on the signal observed during the $2\,\mathrm{ms}$ off period of the device, when $f = 0$. The dynamic noise term is derived according to Equation 3.3.

$$n = \frac{r_0 + b}{g} \tag{3.3}$$

3.3.2 Channel gains

Due to component tolerances, each channel has a slightly different gain. To measure these gains, I place the controller within the Helmholtz coils and rotate it while recording data. In a properly calibrated system, the magnitude of the magnetic field measurement would remain constant as the device is rotated. We use a second optimizer to learn the optimal gains such that the magnitude of the calibrated field remains constant.

Figure 3.8 shows the observed magnitude (blue) and calibrated magnitudes after accounting for the signal chain model (orange), channel gains without the signal chain model

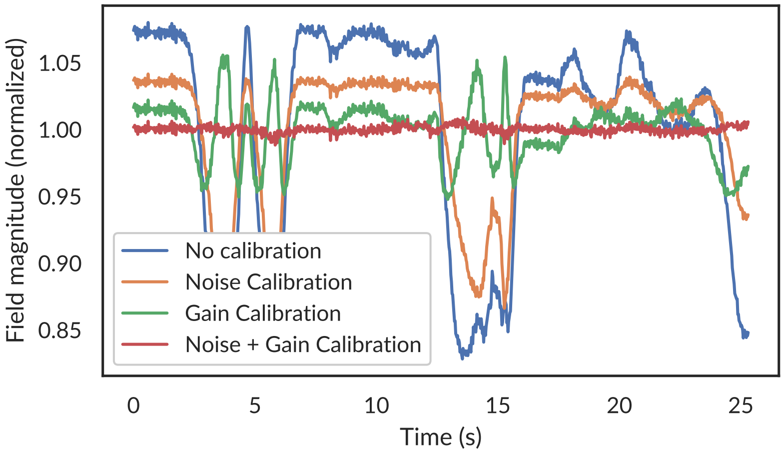


Figure 3.8: Within the Helmholtz coil, the measured field magnitudes should be constant as the controller is rotated. As different calibration steps is applied, the calibrated signal significantly improves.

(green), and both models (red). The constant magnitude obtained after calibration validates these models and shows that both are critical in obtaining correct magnetic field measurements.

3.4 Tracking Algorithm

Because it is infeasible to construct an analytic formulation of the magnetic fields, there is no closed-form analytic solution for the inverse problem. Instead, we rely on a closed-form data-driven solution based on machine learning techniques. Our proposed closed-form tracking approach is summarized in Figure 3.9. We treat the process of position and orientation estimation separately. At a high level, we train a neural network to regress from extracted rotation-invariant features to position. We then train a separate network to estimate the magnetic field vectors from position and use SVD to compute orientation.

Unlike other approaches, this pipeline makes no assumptions about the shape of the

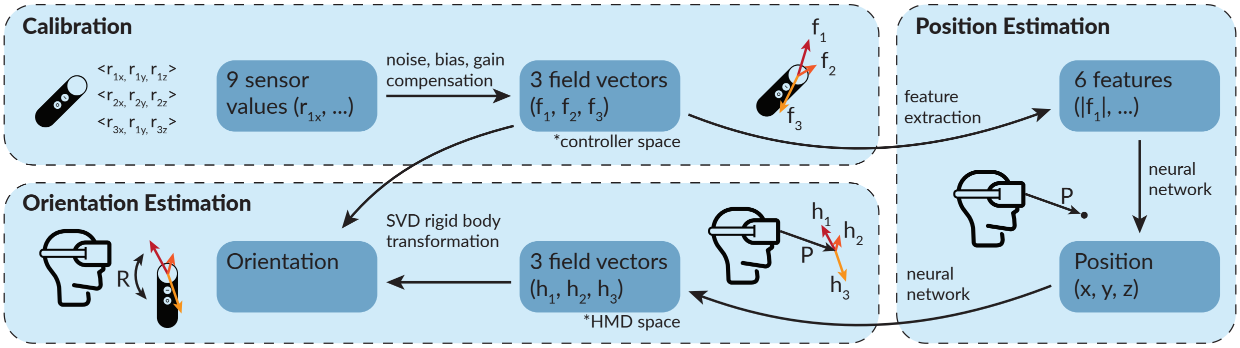


Figure 3.9: Features extracted from the field vectors are first used to estimate position. The estimated position is used to estimate the field vectors in the headset space. SVD is used to extract the orientation from these paired field vectors.

transmit coils or the resulting field (other than it being separable by the features I extract). This frees us from some of the restrictions of a model-based approach, which excels only when the magnetic field model accurately represents the empirical data. The training methods I describe are intended to be a one-time factory calibration.

Aura must estimate the position of the controller from the three measurements of magnetic field vectors (f_1, f_2, f_3). Because the controller can exist at the same position with any rotation, we first extract six features that are invariant to the rotation of the controller. The first three features (Equation 3.4) include the magnitude of all measured fields. As the controller rotates, the *magnitude* of each of the three fields remains constant. The second three features (Equation 3.5) relate to the angle between two fields. As the controller rotates, the direction of each measured field in controller space, f_i will change, but the angle between any two fields will remain constant. Note that in Equation 3.5, I take the absolute value of the dot product. This is because Aura do not have an absolute sign reference between our transmitter and receiver coils. That is, Aura do not know whether an ideal sensor would have measured f_i or $-f_i$. By removing the sign of the dot product, Aura remain invariant

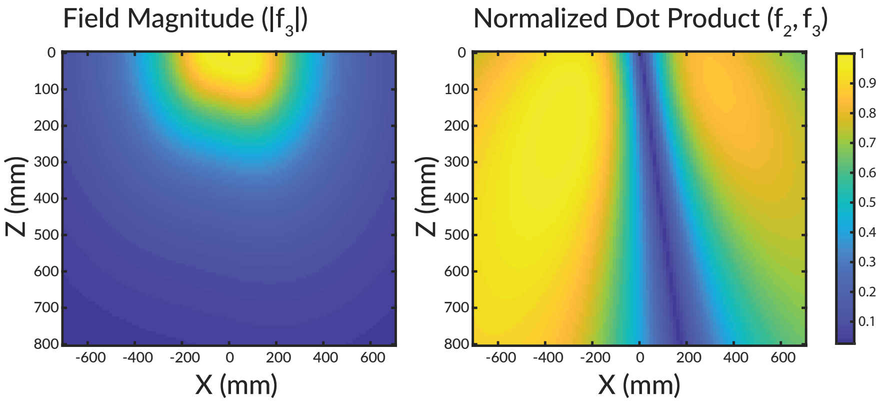


Figure 3.10: A cross-section at y = -300 mm of two of rotation-invariant features from the simulated dipole dataset. The left plot shows the magnitude of the magnetic field from the side one of the side coils, per Equation 3.4, and the right plot shows the dot product between the center and side coil, per Equation 3.5.

to this ambiguity.

$$|f_i|, \forall i \in \{1,2,3\} \tag{3.4}$$

$$\frac{|f_i \cdot f_j|}{|f_i| \times |f_j|}, \forall (i,j) \in \{(1,2),(2,3),(1,3)\} \tag{3.5}$$

These features summarize the relative directions and strengths of the three magnetic fields. A cross-sectional slice of two of these features from the simulated dipole dataset are depicted in Figure 3.10.

3.4.1 Position

We propose the use of computationally simple models to regress to a position vector. In order to keep the models small, we split the tracking volume into four subspaces along two dimensions: 1) left ($x < 0\,\mathrm{mm}$) / right ($x > 0\,\mathrm{mm}$) and 2) near ($|P| < 200\,\mathrm{mm}$) / far

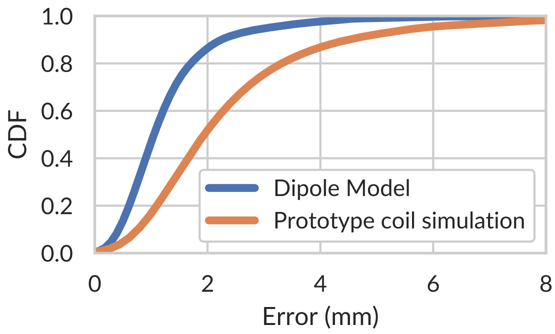


Figure 3.11: CDF of simulated tracking error for the dataset generated using dipole field models and numeric integration of the actual prototype coils.

($|P| > 200\,\mathrm{mm}$). These volumes were empirically determined to balance model performance and complexity. By reducing the tracking volume for each model, I can train much smaller models than I could if the entire tracking volume were lumped together. For each of the four subspaces, I train a computationally simple neural network model with a single hidden layer of 32 nodes to fit a function that maps the six rotationally invariant features to a 3-dimensional position vector. Training is performed using the Levenberg-Marquardt algorithm. At runtime, position estimation equates to two matrix multiplications (6×32 and 32×3), that can easily run on a mobile processor.

It must be noted that this approach adds a dependency on knowing the position before one of these four models can be chosen (to then estimate position). In this work, we use the ground truth position to pick the correct subspace model in order to validate each model separately. However, since the controller will be tracked over time, the next position can easily be estimated from temporal extrapolation or Kalman filter prediction with an IMU. This rough estimate can be used to choose the appropriate model for that frame. Additional redundancy can be added by training intermediate models that straddle two subspaces.

We simulated our approach using magnetic field simulation tools that rely on quasi-static assumptions to generate two datasets. First, we model our transmitter coils as ideal dipole models and use physics calculations to compute the magnetic field due to each of the three coils at many locations around the head. As a comparison, we also use the BSMag toolbox [97] in MATLAB to model the specific shape of our transmitter coils and create a similar dataset of magnetic field locations.

Each dataset consists of 100,000 points generated within a tracking volume of $1.6\,\mathrm{m} \times 0.8\,\mathrm{m} \times 0.8\,\mathrm{m}$. Our model is able to estimate the position of the synthetic datasets with a median error of $1.04\,\mathrm{mm}$ for the dipole model and $1.9\,\mathrm{mm}$ for the simulation of the Aura coil design. Figure 3.11 shows the positional tracking error CDF for simulated data on the dipole model and Aura prototype model.

3.4.2 Orientation

We derive the orientation, or attitude, of the controller by comparing the measured magnetic field vectors in controller-space ($\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$), with estimates of the magnetic field vectors in head-space ($\hat{h}_1$, $\hat{h}_2$, $\hat{h}_3$), derived from a forward model. In some electromagnetic tracking systems, one can compute the magnetic fields ($\hat{h}$) as a function of position using a magnetic field model, often a dipole model [59]. In our system, I anticipate significant deviations from an ideal dipole model due to the presence of electronics around the head, the non-circular nature of our coils, and the arbitrary positioning of the coils for form-factor purposes. Instead, for a forward model, I train a separate neural network to estimate the magnetic field vector at any position, p around the headset. We again adopt a computationally-simple neural network with a single hidden layer of 32 nodes that maps the 3D position, $\hat{p}$, to three 3D vectors, $\hat{h}_1$, $\hat{h}_2$, and $\hat{h}_3$. The model is trained using a ground truth source of position and an estimate of $\hat{h}$ computed by rotating the measured $\hat{f}$ by a ground truth source of orientation. This training process is again intended to be a one-time calibration procedure.

To obtain an orientation estimate at runtime, Aura uses the magnetic field estimates and the well-known singular value decomposition (SVD) method to find the least-squared error between the two coordinate systems. For a complete review of this approach, see

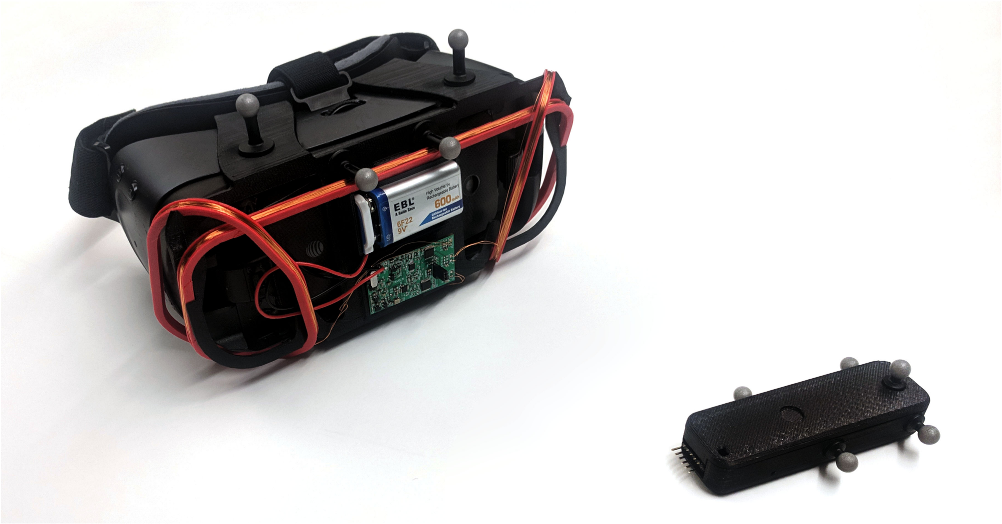


Figure 3.12: Retroreflective markers placed on the HMD and controller enable tracking with a ground truth optical motion capture system.

Sorkine-Hornung and Rabinovich [119].

3.5 Data Collection

3.5.1 Optical Motion Capture

A ground truth source for position and orientation is required to train and evaluate Aura's tracking models. To this end, we use a seven-camera Vicon motion capture system to record the real-time position and orientation of both the headset and the controller at $240\,\mathrm{Hz}$. To enable tracking, we place retroreflective spheres on both devices in known locations, as shown in Figure 3.12.

Importantly, we define the coordinate system of the controller to be the precise center of the magnetic coil so that a rotation about the origin does not change the position of the sensor. We run a one-time calibration step to find the rigid body transform between the object coordinate systems reported by Vicon and our desired coordinate system. We use the average position of the markers from Vicon and the expected marker positions from

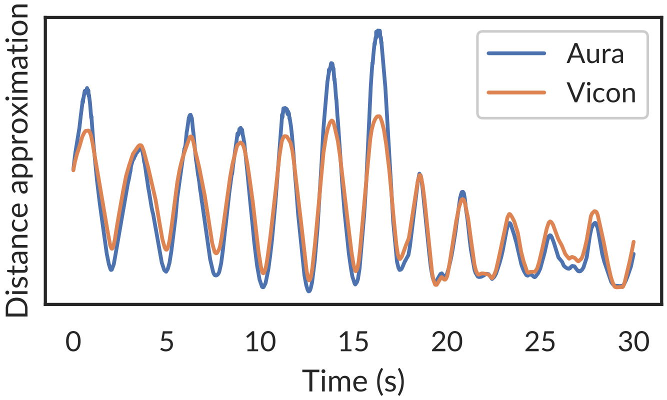


Figure 3.13: Signals used to compute alignment between Aura and the motion capture system.

the known geometry of our system to derive this transformation, which we then apply to each frame that the Vicon system reports.

Because Aura estimates the relative pose of the controller with respect to the head, we use standard coordinate system transformations to compute this 6-DoF pose from the pose of each device in room coordinates.

3.5.2 Synchronization

Once data has been recorded from both the $91\,\mathrm{Hz}$ Aura system and the $240\,\mathrm{Hz}$ motion capture system, we must align and synchronize the two data streams. Without an electrical synchronization signal between the two, we rely on characteristics of each signal for alignment. For this, we compare the distance between the controller and headset as measured from motion capture with an approximation of the distance from Aura as specified by Equation 3.6. While this signal does not have a physically significant value, it was found to correlate strongly with the distance from the headset, as shown in Figure 3.13.

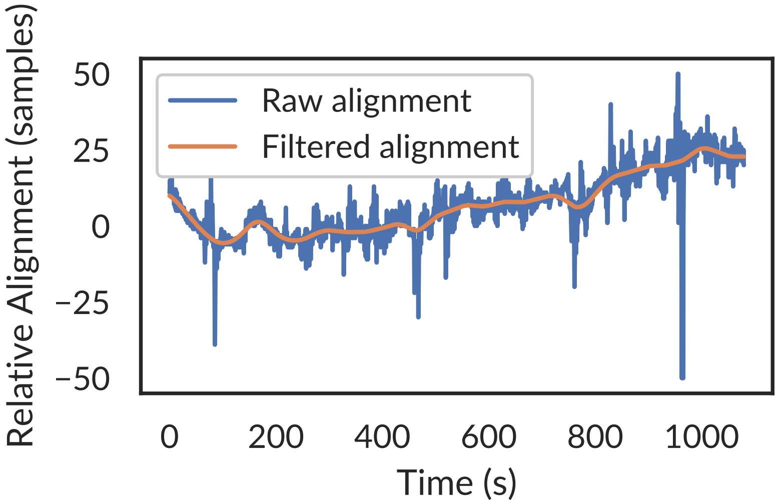


Figure 3.14: Cross-correlation between the sensor data and the distance from the headset.

$$\frac{1}{\sqrt{\sum_{i=1}^{9} f_i^2}} \tag{3.6}$$

We first use these signals to achieve alignment at the start and end of the recorded data streams. However, due to effects like temperature changes, it was observed that the two clocks drift relative to each other over time by as much as 1 part per thousand. At a typical hand speed of $10\,\mathrm{cm/sec}$, each frame misalignment represents an additional error of over $1\,\mathrm{mm}$, so mitigating the effects of this drift is essential.

We use sliding cross-correlation between the two signals to compute a dynamic estimate of frame shift and filter this signal for smoothness. Figure 3.14 shows how the two systems drift over time. We then resample the motion capture data to align with the Aura system.

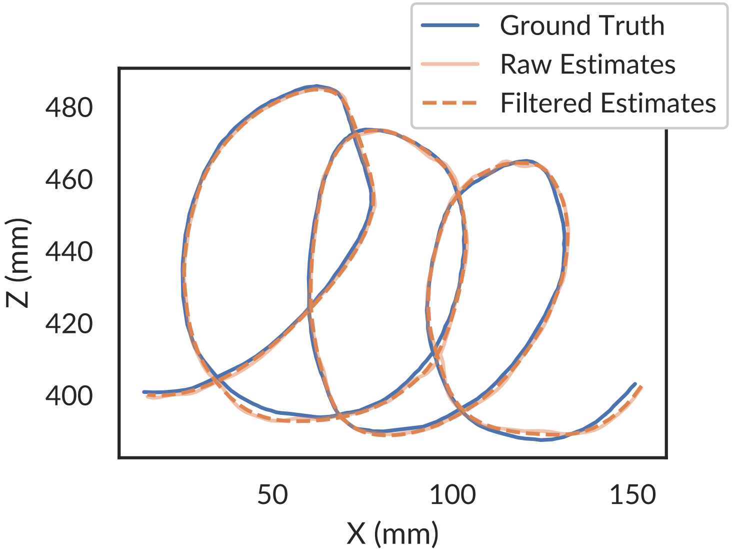


Figure 3.15: 2D positional tracking performance

3.6 *Evaluation*

The main performance metrics for Aura are position and orientation accuracy, precision analysis, latency, and power consumption. We also evaluated the effects of magnetic interference on Aura's measurements.

3.6.1 Position Estimation Accuracy

2D Position Accuracy

As an initial verification of the tracking capabilities of Aura, we evaluated its positional tracking accuracy in a constrained 2D task. For this experiment, the headset was placed on a flat surface while the controller was manually moved along the surface at a mean distance of $0.5\,\mathrm{m}$ from headset. First, training data was collected by sweeping the controller across a $44\,\mathrm{cm} \times 32\,\mathrm{cm}$ area. A test set was collected by randomly moving the controller about within the same tracking area.

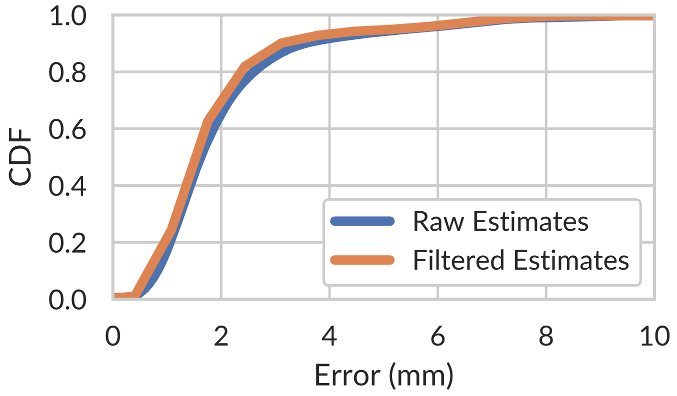


Figure 3.16: CDF of 2D positional error

Figure 3.15 shows the trace from the Vicon data and the reconstructed path from Aura. The Aura system is able to track the controller with a 2D median tracking error of $1.6\,\mathrm{mm}$. A Kalman filter is used to smooth the estimate of position and reduces the median error to $1.5\,\mathrm{mm}$. Figure 3.16 shows the CDF of 2D position accuracy for the raw and filtered estimated of position.

3D Position Accuracy

While the 2D test demonstrates the feasibility of this approach, Aura is intended to track handheld controllers in 3D space. For evaluating the 3D positional tracking capabilities, one of the authors wore the headset while holding the handheld controller and moving it about within arm's reach. Training data was collected for $15\,\mathrm{min}$ by systematically exploring the space in front of the user. An additional two minutes of random motion was then collected as the test set.

Figure 3.17 shows the ground truth and estimated position over a representative $30\,\mathrm{s}$ segment of the test set. Aura is able to track the controller with a 3D median error of $7.0\,\mathrm{mm}$. After applying the Kalman filter, the median error drops to $5.5\,\mathrm{mm}$. Figure 3.18

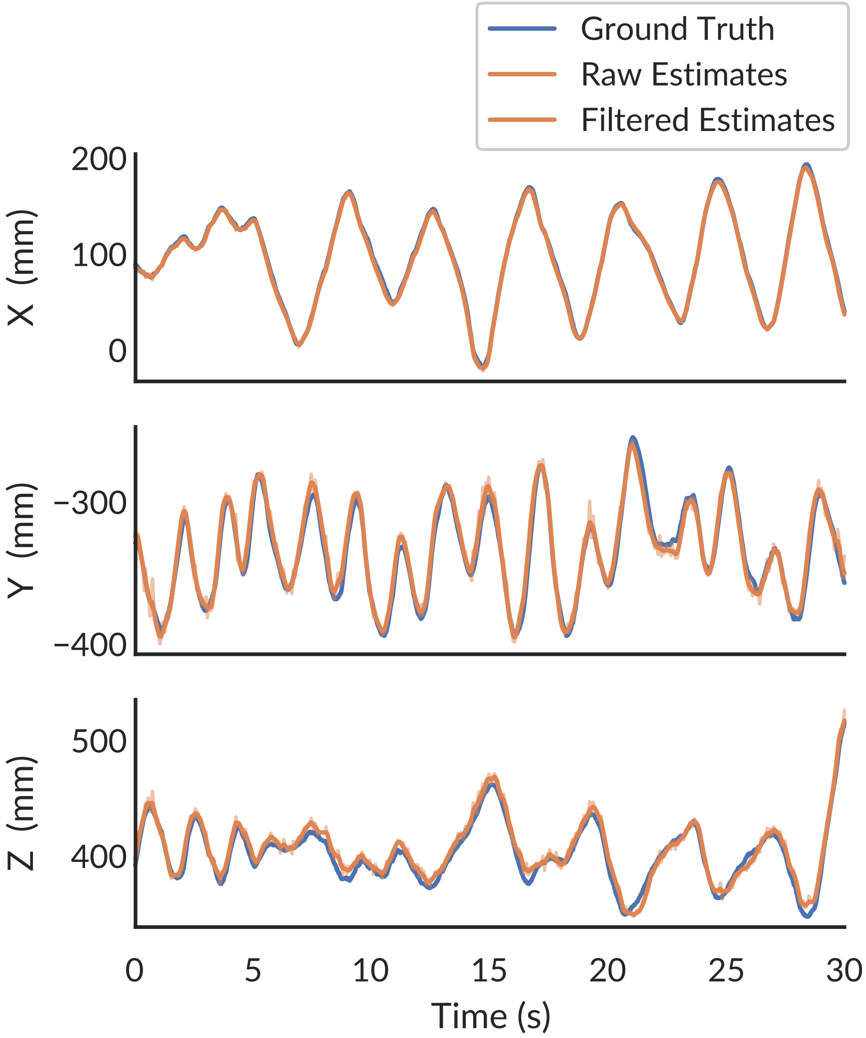


Figure 3.17: 3D positional tracking performance

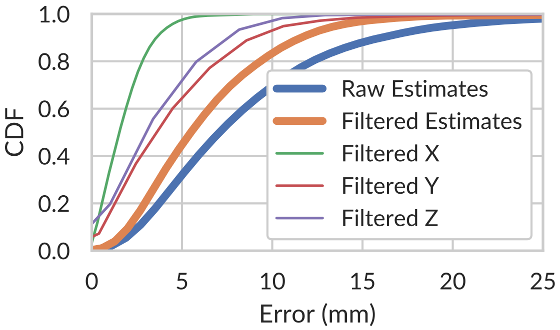


Figure 3.18: CDF of 3D positional error

shows the CDF of 3D position accuracy for the raw and filtered estimates of position. We expect further performance gains by fusing the electromagnetic tracking with an onboard IMU.

3.6.2 Orientation Accuracy

We use the same dataset to train the magnetic field models for orientation estimation. Aura estimates the orientation of the handheld controller using the algorithm described in Section 3.4 and reached a median accuracy of $0.8°$. Once again, we expect significant performance gains after fusing this data source with an IMU. However, as an initial approximation, we simulate the effects of leveraging an accelerometer to recover the gravity vector. We derive this estimate of the gravity vector from the ground truth motion capture system. Adding this additional vector reference to the SVD calculation improves the median orientation accuracy to $0.5°$. Figure 3.19 shows the CDF of orientation accuracy for both Aura and the simulated approach using the added gravity vector.

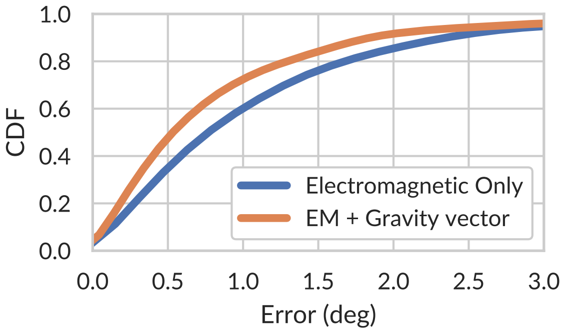


Figure 3.19: CDF of orientation tracking error

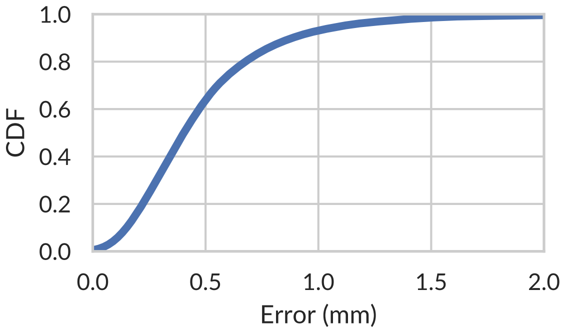


Figure 3.20: CDF of stationary measurement jitter

3.6.3 Precision Analysis

We also evaluated Aura's precision in its tracking estimates. To measure the precision, we used the same set up as in Section 3.6.1. However, instead of continuous motion, the device was placed in eight different locations and left motionless for a few seconds. We then compute the jitter in the estimated position while the device is motionless. The median jitter across all points is calculated to be $0.4\,\mathrm{mm}$. This calculation includes the digital low-pass filter applied to the raw magnetic signals, but no additional Kalman filtering of the position. Figure 3.20 shows the CDF of jitter.

3.6.4 *Speed and Latency*

Although the tracking analysis was performed offline, it was designed with realtime operation in mind. Like any handheld controller, we envision Aura to be used alongside an onboard IMU to capture high-speed motions. Nonetheless, we characterize the latency and speed aspects of our system.

Aura's latency is impacted by delays introduced by the analog signal chain, digital signal processing, and tracking algorithm. To quantify the impact of the analog signal chain, we place the Aura controller within the Helmholtz coil and activate the field while measuring both the current through the Helmholtz coil and the input the to Aura ADC on a oscilloscope. The time between the field turning on and the signal stabilizing is $200\,\mu\mathrm{s}$. This introduces negligible latency and validates our decision to use $3\,\mathrm{ms}$ "on" states for each coil. On the digital side, the use of interpolation to resample the magnetic signal reduces the impact of the $3\,\mathrm{ms}$ delay between channels. An additional digital filter is used to smooth data and can be set according to anticipated device usage. In our prototype, a second-order Butterworth filter with a $10\,\mathrm{Hz}$ cutoff frequency was used. Finally, we anticipate negligible latency from the tracking algorithm, which was designed with simple computation in mind. The computation consists primarily of a few small matrix multiplications, which can be performed in real time on most microprocessors.

3.6.5 *Power*

We measured the power consumption of both the Aura transmitter and receiver. Our measurement setup consists of a National Instruments (NI) USB 6003 data-acquisition (DAQ) unit configured for taking analog measurements in fully differential mode $-10\,\mathrm{V}$ to $10\,\mathrm{V}$ at a sampling rate of $10\,\mathrm{kHz}$. We measure the voltage drop across a shunt resistor of size $10\,\Omega$ for amount of $5\,\mathrm{s}$ and calculate the power consumed in Aura.

The handheld controller consumes an average of $13.5\,\mathrm{mA}$ ($45\,\mathrm{mW}$. Using a $700\,\mathrm{mA\,h}$ LiPo, which would comfortably fit in the controller housing, the Aura handheld controller could be tracked continuously for over two full days. Note that this does not include power consumption for a Bluetooth module or other wireless communication device.

The head-mounted transmitter consumes $29.8\,\mathrm{mA}$ average current ($224\,\mathrm{mW}$). Using a $9\,\mathrm{V}$ battery ($600\,\mathrm{mA\,h}$), the transmitter lasts for $24\,\mathrm{h}$ of continuous use. Although the current design of the transmitter uses a $9\,\mathrm{V}$ battery to operate, the Aura design is easily modified to operate at $3.3\,\mathrm{V}$. Adding more windings to the transmitter coils and reducing the operating voltage has led to prototype designs that consume only $30\,\mathrm{mW}$ on the head-mounted transmitter with similar field strengths. With this design, the Aura transmitter would last nearly four days on a $700\,\mathrm{mA\,h}$ LiPo battery.

For reference, the electromagnetic Polhemus G4 tracking system uses $5\,\mathrm{W}$ for the transmitter and $2.5\,\mathrm{W}$ for the sensor hub [96]. The HTC Vive Lighthouse base stations, an optical head and controller tracking solution, use approximately $5\,\mathrm{W}$ each, as measured by an inline power monitor. The Magic Leap One controller contains an $8.4\,\mathrm{W\,h}$ battery and is rated for $7.5\,\mathrm{h}$ of continuous use [66], suggesting a power consumption on the order of $1\,\mathrm{W}$. Aura uses at least an order of magnitude less power than these alternatives.

3.6.6 Interference

Fundamentally, electromagnetic tracking is prone to interference from nearby metallic objects. While this work does not claim any specific algorithmic contributions to account for interference, we note that the use of inside-out tracking, as opposed to outside-in tracking, significantly reduces the scope of possible interference sources. Based on our observations, significant distortion was only observed when metallic objects were placed close to either the transmitter or receiver. To quantify these effects on the Aura system, we investigated the interference from three common electronic devices that are likely to be in close proximity to the system: a smartwatch, a smartphone, and a laptop. The Aura controller and headset are placed on a flat surface as in 3.6.1. The authors then bring each interfering device from a far distance to close proximity to either the handheld controller

Table 3.1: Effects of various devices on the Aura sensor readings. Each distance represents the closest distance at which the device changes the measured signal by less than 1%.

Device	Distance to transmitter	Distance to receiver
Smartwatch (Apple Watch Series 2)	2 cm	2 cm
Smartphone (iPhone 7)	10 cm	5 cm
Laptop (2018 MacBook Pro)	20 cm	20 cm

or the HMD while recording the change on the received signals. The distance at which the signals changed by 1% was recorded. The results from this experiment are summarized in Table 3.1.

In summary, a small electronic device such as a smartwatch, has no effect as long as it is at least a few centimeters from the Aura system. Larger electronic devices must be kept further away before significant distortion is observed. As with any other electromagnetic tracking system, larger ferromagnetic materials, such as iron beams or vehicles, would have a much larger impact on the signal. While accounting for such distortions is out of scope for this work, we note that there is existing research on accounting for sources of interference [61, 60]. We also note the possibility of fusing electromagnetic tracking with inertial or optical tracking to dynamically calibrate in the presence of dynamic sources of interference.

Although not common today, it is important to consider the implications of using multiple devices within the same room. In theory, the signal from one Aura device could interfere with another if they are tuned to the same frequency. Fortunately, the strength of the generated magnetic field falls off with the cube of the distance to the transmitter. We measured the distance from the headset at which the signals fall below the noise floor of the sensor to be $1.5\,\mathrm{m}$. This indicates that there will be negligible interference between separate systems as long as they remain more than $1.5\,\mathrm{m}$ apart. For optimal operation at closer distances, the two devices should be set to different frequencies to avoid interfering.

3.7 Conclusion

In this work, we present Aura, a head-mounted inside-out tracking system for handheld devices. We demonstrate a novel low-power architecture that enables precise tracking without the need for external infrastructure, line-of-sight, or bulky tracking markers. In an evaluation with an optical motion capture system, we demonstrate Aura's ability to track a controller with a median error of $5.5\,\mathrm{mm}$ and 0.8° within arm's reach. We hope that our system enables increased adoption of mobile spatial computing systems.

Chapter 4

PRECISE FINGER TRACKING

4.1 Introduction

New computing platforms are becoming increasingly coupled to the user. Wearable devices like smartwatches, smart rings, and head-mounted mixed reality devices are offering unprecedented access to computing and information on the go. As computing platforms evolve from devices we carry with us to devices we wear on us, there is a renewed demand for input techniques that are decoupled from the display. These input devices must be continuously available and subtle to support various contexts of use while enabling robust and expressive interaction.

In this chapter, we present AuraRing, a precise, millimeter-level finger tracking system. AuraRing consists of two components: a ring and a wristband. The low-power battery-operated ring generates an oscillating magnetic field around the hand. As the user moves their finger and wrist, the relative position and orientation of this field changes. Sensors embedded in the wristband measure these fields at different locations and estimate the pose of the ring with respect to the wristband. By using oscillating magnetic fields, AuraRing is not affected by the Earth's geomagnetic field. AuraRing is effectively a 5 degree-of-freedom (DoF) tracking system—that is, it tracks three positional components and two rotational components (yaw and pitch) of pose. This allows AuraRing to capture flexion/extension and abduction/adduction of both the wrist and finger metacarpal joint. Although the ring can be worn on any finger, it is intended to be worn on the index finger for pointing tasks.

Compared to prior work, our approach leverages common device form-factors—a wristband and a ring—and does not require affixing magnets [44, 15, 49, 14], iron-core coils [16], or magnetometers [15, 49, 14] to the fingertips. AuraRing leverages the insight that an elec-

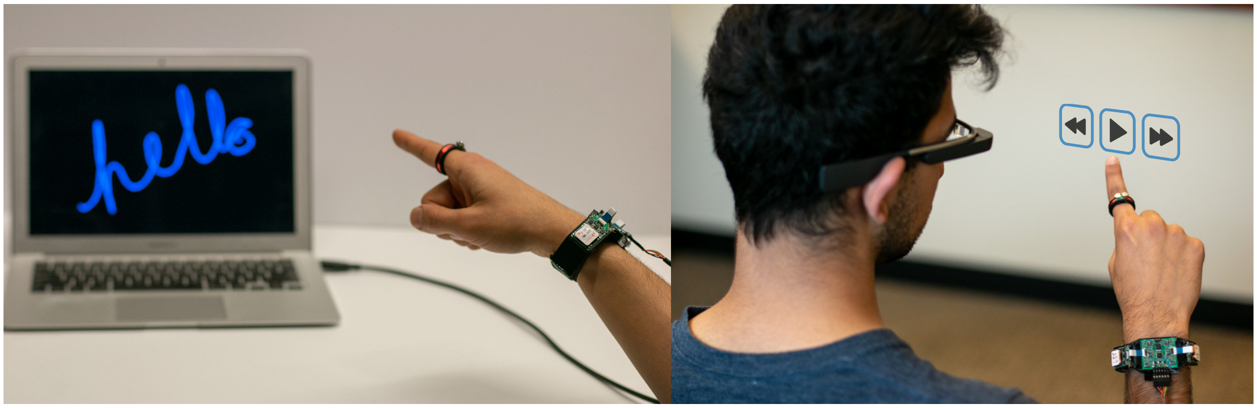


Figure 4.1: AuraRing is 5-DoF electromagnetic tracker that enables precise, accurate, and fine-grained finger tracking for AR, VR and wearable applications. Left: Show a user writing the word ”hello” in the air. Right: Person using AuraRing to play a song in a music application on a smart glass.

tromagnetic coil is better placed around the finger than on it. Our approach also delivers significant improvements in power, range, portability, and tracking precision.

Electromagnetic tracking has a long history of use in meter-scale tracking applications [62, 114, 116, 91, 85, 96] in a variety of domains. Though precise, electromagnetic tracking has a reputation for being power-hungry and subject to environmental interference. AuraRing overcomes these challenges by explicitly focusing on short-range ($10\,\mathrm{cm}$ to $15\,\mathrm{cm}$) tracking. This range reduces the scope of possible interfering objects and allows the AuraRing transmitter to use approximately 2000x less power than a typical commercial EM tracking system [96].

Sensor-based efforts to track finger motion in space often focus on discrete, gesture-based interactions [37, 137, 102, 159, 7]. Although gesture recognition is useful, wearable input devices must be robust and support different users and contexts of use. To enable this level of precision and robustness, AuraRing supports high-fidelity continuous tracking, upon which gesture recognition or other interactive systems can be built. Because AuraRing leverages physics models, it works out of the box after a one-time factory calibration with ground-truth tracking data from a motion-capture system. We demonstrate that at runtime, AuraRing can track finger motion across different users and despite changes in how the device is positioned on the wrist and finger. In cases where additional accuracy is desired, the calibration can be improved with just $60\,\mathrm{sec}$ of additional data.

AuraRing offers a flexible solution for tracking the finger in either direct or indirect pointing tasks. We envision that AuraRing can be used either as a standalone input device, or in tandem with a wrist-tracking solution. For standalone scenarios, AuraRing offers a rich input source for smartwatches or smart glasses. With AuraRing, a user can provide input using their finger and wrist while keeping their arm motionless at their side or resting on a table. This capability is particularly useful in public settings, when large hand motions would be distracting and socially unacceptable.

AuraRing can also be used to track the absolute position of the finger in head-space with respect to the user's vision. In mixed-reality scenarios, precise hand and finger tracking enables many compelling applications. For example, buttons, sliders, and interactive widgets can be placed in the air, on environmental surfaces, or on the body. AuraRing can be operated in concert with a head-mounted camera that tracks the wrist with respect to the head using fiducials on the wristband.

Our primary contributions include:

1. A hardware architecture for a ring and wristband that enables low-power finger tracking in a compact form-factor.

2. Two tracking algorithms, including a physics-based iterative approach and a closed-form neural network approach to 5-DoF pose estimation.

3. A system characterization and user evaluation demonstrating mean tracking a resolution of $100\,\mathrm{\mu m}$ and dynamic accuracy of $4.4\,\mathrm{mm}$ on a session-independent task.

4.2 Theory of Operation

4.2.1 AuraRing Physics

Magnetic tracking is generally realized either with magnetometers to track a permanent magnet [7, 15, 49] or with inductive coils to track an alternating current (AC) electromagnet [114, 116, 143, 147, 96, 85, 98, 62]. Static magnetic fields created by a permanent magnet or inseparable from the Earth's geomagnetic field—an effect which becomes critical when operating at distances beyond a few centimeters. To isolate the signal of interest, AuraRing uses AC electromagnetic coils to generate a magnetic field at a particular frequency ($32\,\mathrm{kHz}$). Maxwell's equations state that an AC magnetic field is generated when an AC electric current is passed through a wire coil. As depicted in Figure 4.2, AuraRing uses a wire coil wrapped on the ring to produce an AC magnetic field around the hand. According to Faraday's law, a voltage is induced in the other sensor coils in the presence of an AC magnetic field. The induced voltage is proportional to rate of change of magnetic flux through the coil. For a field generated from the and electromagnetic, the magnetic flux and resulting voltage signal will change as a function of the position and orientation of the sensor coils in respect to the transmitter coil. More specifically, if the sensor coil is aligned with the field (i.e., the normal vector to the coil is aligned with the field), then the magnitude of the induced voltage will be maximized. As the coil rotates away from the field, the induced voltage decreases to zero. If the coil continues to rotate even further, the voltage acquires a $180°$ phase shift that manifests as a negative amplitude.

Traditional 6-DoF electromagnetic trackers use a 3-axis transmitter and a 3-axis sensor coil, which is difficult to achieve in a ring form-factor where size and battery life are paramount. However, a ring is a convenient form-factor for a single-axis air core coil that wraps around the finger. Using a single-axis coil makes the entire system insensitive to changes in the roll of the transmitter along the magnetic axes; conveniently, such movement is not physically possible in finger-tracking scenarios. To compensate for the lack of three transmitter axes, AuraRing leverages three 3-axis sensor coils embedded at known

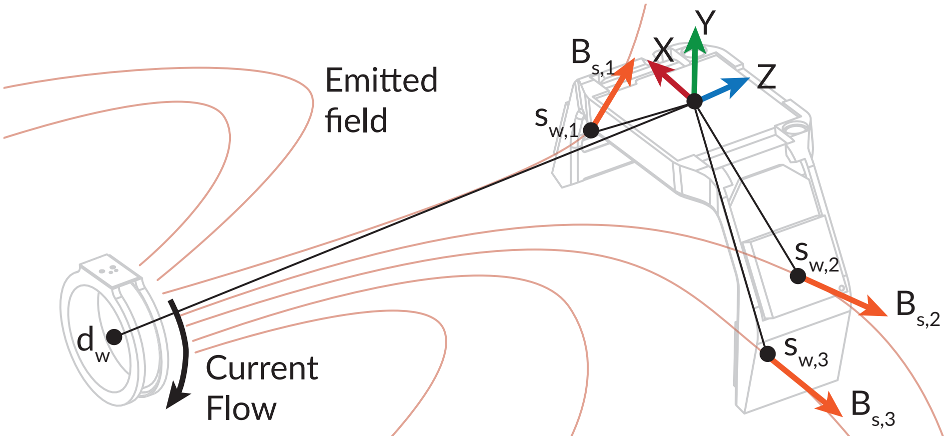


Figure 4.2: AuraRing uses a wire coil wrapped on a ring to produce an AC magnetic field around the hand which is measured by three 3-axis coils embedded in a wristband.

locations within a wristband. By measuring the magnetic fields at different points in space, AuraRing reconstructs the 3-DoF position and 2-DoF orientation of the ring. Figure 4.2 illustrates the configuration of the AuraRing transmitter and sensors.

4.2.2 Magnetic Field Model

We construct a physics-based simulation to model the behavior of our system under different configurations, such as changing the number or placement of sensor coils. Because the ring diameter is much smaller compared to the distance between the ring and wristband, we use standard magnetic field equations to model the ring as a dipole emitter. The model seeks to estimate the magnetic field ($\vec{B}_s$) measured at each sensor coil (s) in the wristband. Because the transmitter is embedded within the object to be tracked, we first conduct a series of coordinate system transformations. In this paper, we adopt the notation $\vec{B}_s$ to represent the $\vec{B}$ vector in the s coordinate frame. We represent a rotation using $\mathbf{s}_w$ to represent a transformation from the w frame to the s frame. In practice, these are implemented using quaternions, but other representations would be appropriate as well.

Let d be the dipole coordinate frame with the magnetic axis oriented in the z-direction.

This is coincident with the ring coordinate frame. Let w be the wrist coordinate frame, which contains multiple sensors s_i at positions $\vec{s}_{w,i}$ and orientations $\mathrm{s}_{w,i}$, as depicted in Figure 4.2. Given a position of the ring ($\vec{d}_w$) and the geometry of the sensors ($\vec{s}_w$), we first compute the position of the sensor in the dipole frame:

$$\vec{s}_d = \mathbf{d}_w \times \left(\vec{s}_w - \vec{d}_w\right) \tag{4.1}$$

We then use standard magnetic dipole equations derived from the Biot-Savart law under quasistatic assumptions to compute the magnetic field at each sensor location ($\vec{B}_{d,i}$):

$$B_x = \frac{3xz}{r^5} \qquad B_y = \frac{3yz}{r^5} \qquad B_z = \frac{3z^2 - r^2}{r^5} \tag{4.2}$$

Finally, we rotate once more to compute the magnetic field in the sensor frame of reference:

$$\vec{B}_s = \mathbf{d}_w^* \times \vec{B}_d \tag{4.3}$$

4.2.3 Number of Sensors Required

To inform the design of AuraRing, we conducted a preliminary analysis using this model in which we varied the number of sensor coils. Choosing a sensor coil configurations requires balancing performance versus size and power. Tracking the ring is at least a 4-DoF task, since both the wrist and finger metacarpal joints have two degrees of freedom. As a result, a robust approach requires at minimum two coils; each coil measures a three-dimensional field vector from a single emitter, so two coils would provide six sensor values. In practice, due to the potential for small shifts of the band on the wrist, AuraRing tracks as a 5-DoF task (x, y, z, pitch, yaw), which we require hypothesized might be difficult with just two sensor coils. A configuration is considered robust if, for a set of observed sensor values, there is a unique position and orientation within reasonable bounds that explains those observations.

To explore the design space of coils, we first choose a test point, representative of a typical ring position and orientation, and use the physics model described previously to compute the magnetic field as observed from each sensor. We then compare this observation to a dense sampling of positions and orientations. The x, y, and z axes are defined according to the wrist frame in Figure 4.2, with the finger generally pointing along the $-Z$ axis. For orientation, we define θ as pitch (in the extension/flexion direction) and ϕ as yaw (in the adduction/abduction direction) For position, we sample at $2\,\mathrm{mm}$ intervals within the bounds of $-40\,\mathrm{mm} < x < 40\,\mathrm{mm}$, $-130\,\mathrm{mm} < y < 80\,\mathrm{mm}$, and $-140\,\mathrm{mm} < z < -50\,\mathrm{mm}$. For orientation we use pitch, θ and yaw, ϕ, and Euler angles to generate a quaternion orientation at 2° intervals within the bounds $-90^\circ < \theta < 90^\circ$ and $-60^\circ < \phi < 60^\circ$. These bounds correspond to typical bounds of the ring for typical bone lengths and range of motion.

Figure 4.3 illustrates an example result of this analysis for a single target point. The points in each plot indicate locations that can produce a set of sensor measurements similar to that produced at the target point under any tested rotation. This means that for a given point in the plot, there is some ring orientation at that position that could be confused for the target point, shown in red. The color indicates the measure of similarity to the target point.

As expected for a device with only a single sensor, there is a large confusion volume along all dimensions. Interestingly, there is a defined minimum for a 2-sensor setup, but the problem is ill-conditioned along some directions. This suggests that a 2-sensor device might suffice with the proper constraints, but any noise could result in significant errors. The 3-sensor setup produces a well-conditioned minimum, justifying our use of three coils around the wristband. Although this particular example does not reveal much about optimal sensor placements, robustness is maximized when the sensor coils placements are maximally spread apart.

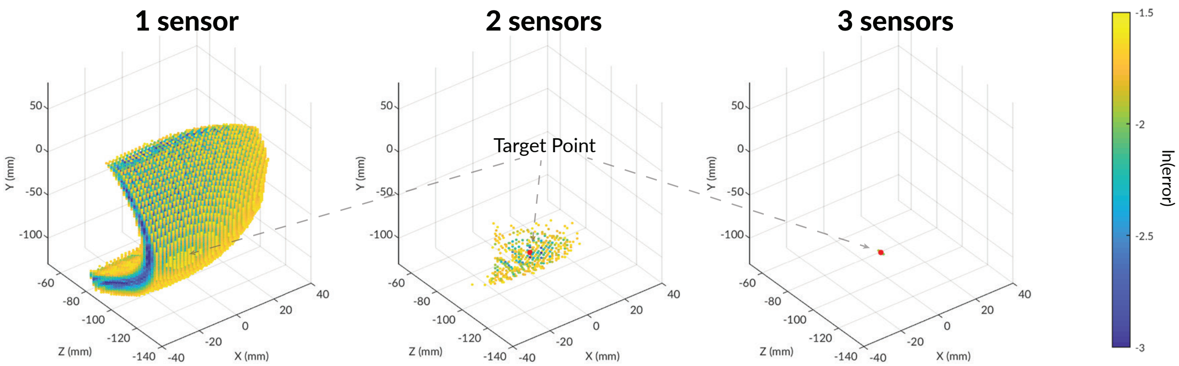


Figure 4.3: By embedding three sensor coils in the wristband, AuraRing achieves a well-conditioned minimum that makes AuraRing robust. As depicted, having one or two coils will result in confusing other points with the target point.

4.3 AuraRing Hardware

The AuraRing system consists of a ring-worn device that generates an AC magnetic field and a wristband device with three embedded sensors that measures the resulting fields. Figure 4.4 depicts the AuraRing system components. The following sections describes the AuraRing hardware, explores the hardware's capabilities, and enumerates design challenges.

4.3.1 Ring

In designing AuraRing, we pursued a design that would resemble device form-factors that users are already accustomed to in order to present a plausible path forward for everyday use. For the ring-based transmitter, rather than affixing an iron-core coil [16] or permanent magnet [14, 15] to the top of the finger, we designed a custom electromagnet transmitter compatible with a ring form-factor. In doing so, we were forced to put minimal digital electronics on the ring to ensure a slim form-factor and minimize power consumption. AuraRing's ring consists of a single-axis low-profile transmitter coil that emits a magnetic field oscillating at $32.768\,\mathrm{kHz}$. The generator coils consists of 800 turns of 42 AWG magnet wire

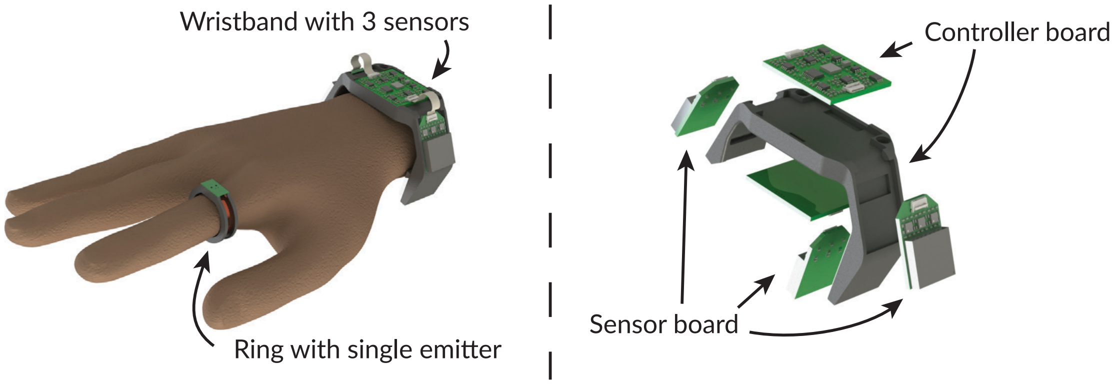


Figure 4.4: AuraRing contains two controller and three sensor boards embedded in a wristband and a ring worn device.

wound around a 3D-printed ABS ring with a diameter of $20\,\mathrm{mm}$. The inductance of the coils is approximately $15\,\mathrm{mH}$. The number of turns and wire gauge was chosen to maximize the magnetic field strength while minimizing DC resistance, skin effects, and weight.

For the transmitter electronics, a small custom PCB ($6.9\ \mathrm{mm} \times 11.4\ \mathrm{mm}$), sits horizontally on top of the 3D printed ring. Figure 4.5 (right) shows a block diagram of AuraRing's transmitter. The transmitter uses a surface-mounted oscillator (ACZ-32.768) to generate a $32.768\,\mathrm{kHz}$ square wave. This frequency was chosen because it is a commonly used frequency for low-frequency clocks on microcontrollers and there is an abundance of integrated chips that produce this frequency. We use a capacitor network to tune the coil's impedance at $32.768\,\mathrm{kHz}$ in order to maximize the transmit power of AuraRing. In short, the series capacitor stores charge that can be drawn through the inductive coil on each cycle of the waveform. This effectively reduces the operating voltage required to generate a given field strength. For a detailed discussion of the effect of these resonant networks on tracking systems, see [52]. The transmitter is powered with two $12\,\mathrm{mA\,h}$, $1.55\,\mathrm{V}$ coin cell batteries with a diameter of $4.8\,\mathrm{mm}$ and a height of $2.1\,\mathrm{mm}$.

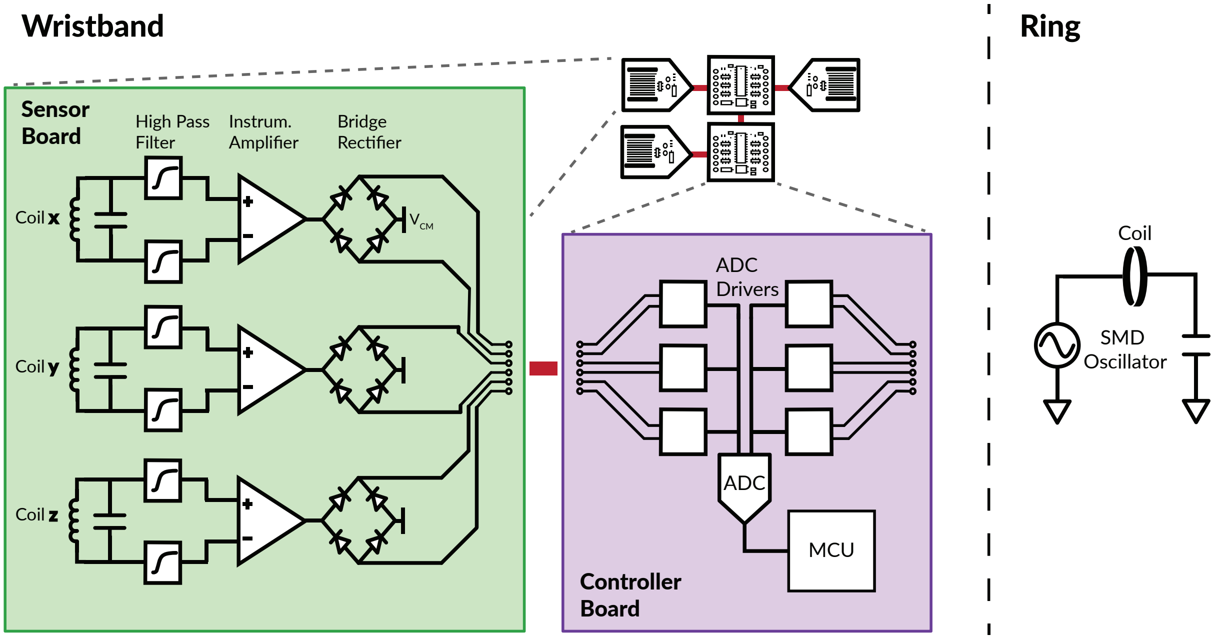


Figure 4.5: Block diagram of AuraRing. The ring generates a magnetic field at a particular frequency which is measured by the sensor boards. They send these signals to the controller board where where an ADC samples the data and communicates with a host computer using an ultra low power MCU.

4.3.2 Wristband

The AuraRingwristband consist of three sensors and two controller boards as depicted in Figure 4.5 (left). At a high level, the sensor boards each measure a magnetic field while the controller boards convert these measurements to digital signals and communicate them to a host computer.

The sensor boards measure the magnetic field generated by the ring using an off-the-shelf three-axis orthogonal receiver coil (Grupo Premo 3DC15). The signal from each axis is fed to an amplifier (INA826) with a gain of $44\,\mathrm{dB}$. We then use a low-noise and low-voltage drop Schottky diode network (SMS7630) in a full-wave bridge rectifier configuration to demodulate each of the channels. We have used this passive configuration to demodulate the field in order to save power. It is worth mentioning that using this method, we only have access to the fields' magnitudes; AuraRing does not know whether each channel is in- or out-of-phase with the transmitter. Because of the hand's kinematics, most of the channels' signs remain constant, so there is not much information lost. In Section 4.5, we show that AuraRing can still track without knowing the exact signs of the fields.

These magnitude signals are passed to a controller board with a 10-wire FFC cable. Each controller board supports two sensor boards. By vertically stacking two controller boards within the center of the wristband, AuraRing supports up to four sensors. Based on the analysis in Section 4.2.3, three coils are sufficient for tracking, so we connect two of the sensors boards to the top controller board and one to the bottom board.

On each of the controller boards, there is an SAR analog-to-digital converter (AD7265) that samples six differential signals simultaneously at $31.25\,\mathrm{kHz}$. Each channel on the controller board has an ADC driver (LTC6363) to achieve high precision, resolution, and throughput. Finally, the sampled data is passed to an ultra-low power MCU (MSP430FR2422) for further processing and communication. Every eight ADC readings, the bottom controller board sends the sampled data to the top controller board where the data is collected and sent to a host computer over USB at $472\,\mathrm{kHz}$. Figure 4.5 summarizes the analog signal processing of AuraRing.

4.4 Calibration

AuraRing's tracking algorithms have, at their core, a physics model, described in Section 4.2.2. This can used to generate synthetic data for model training. However, before the physics model can accurately describe the AuraRing measurements, we must learn a number of parameters, such as channel gains and precise sensor positions. In a precise manufacturing environment, these parameters can likely be set based on an analysis of the system geometry and electronics. However, in our prototype, due to the tolerances of 3D printed parts and our selection of electronic components, we opt to learn these parameters empirically

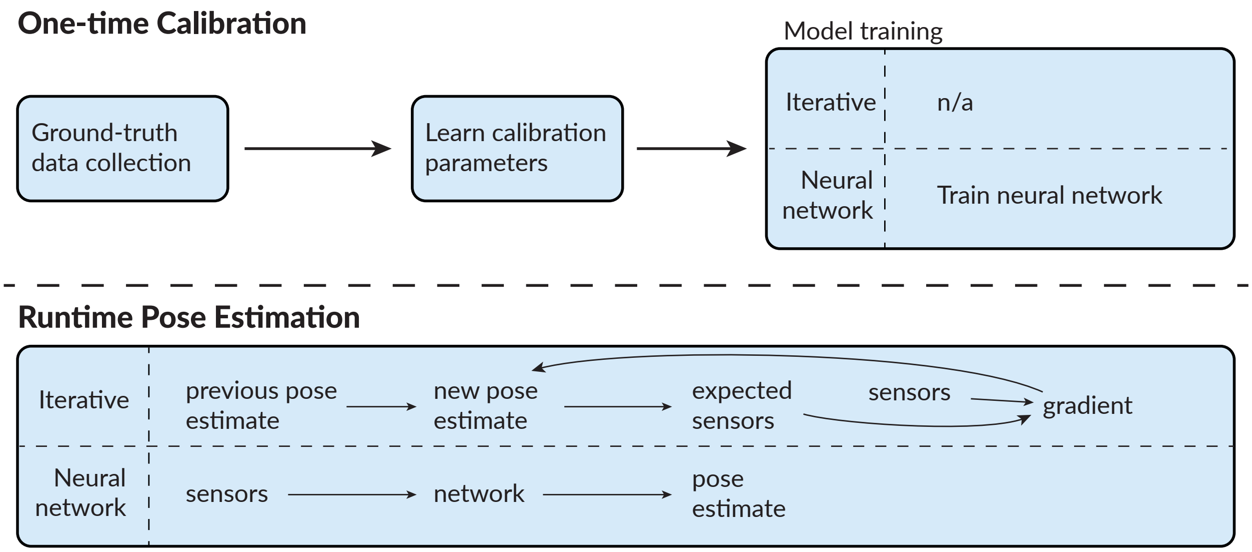


Figure 4.6: The AuraRing system needs a one-time factory calibration to match the synthetic data to its measurements. In real-time, the iterative model uses this calibration to refine the pose estimation and the neural network model regresses directly to pose.

by collecting and using data from an optical motion capture system. In the following sections, we describe our data collection setup, the calibration model, and learning procedure. Figure 4.6 shows a high-level overview of the calibration process and how it relates to the runtime pose estimation task.

4.4.1 Data Collection Setup

We use a 10-camera[1] optical motion capture system (calibrated accuracy of $0.1\,\mathrm{mm}$) to track the ground truth position and orientation of the wrist and ring. To facilitate this, we place IR retroreflective markers on both devices, as shown in Figure 4.7. The motion capture system reports the pose of the wristband ($\vec{w}_m, \mathbf{w}_m$) and ring ($\vec{r}_m, \mathbf{r}_m$) in room-space at $240\,\mathrm{Hz}$. Software in Python on a PC logs both the motion capture data and AuraRing

---

[1] OptiTrack Prime 13

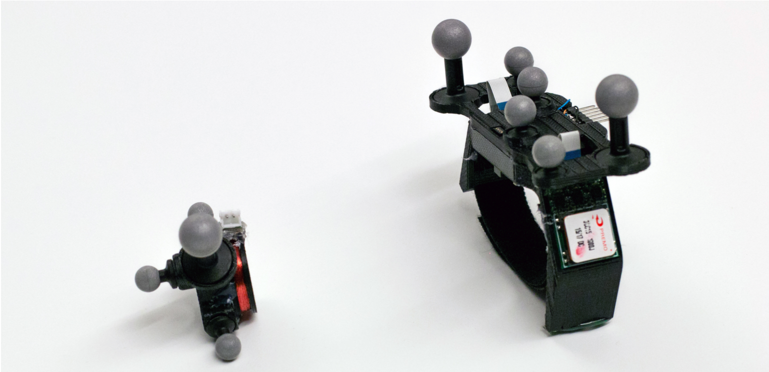


Figure 4.7: We place retromarkers on the ring and wristband to get the ground truth measurement from an optical capture system.

sensor data to disk for offline analysis.

We first preprocess all of the motion capture data to align the rigid body coordinate space to our coordinate system defined by the device geometry, as shown in Figure 4.2. Next, to reconstruct the relative pose of the ring in wrist space, we apply another coordinate system transformation.

$$\vec{r}_w = \mathbf{w}_m \times (\vec{r}_m - \vec{w}_m) \tag{4.4}$$

$$\mathbf{r}_w = \mathbf{r}_m \times \mathbf{w}_m^* \tag{4.5}$$

Finally, we synchronize the sensor stream from AuraRing ($472\,\mathrm{Hz}$) with the ground truth ring pose stream ($240\,\mathrm{Hz}$). We align the two streams by using the ring pose to roughly estimate the magnetic fields, as described in Section 4.2.2, and comparing these fields to the AuraRing sensor values. We then resample the AuraRing signal through interpolation to $240\,\mathrm{Hz}$, align with the ground truth signal. During this process, we drop any frames in which the motion capture system lost track of either the ring or wristband.

4.4.2 Sensor Model Parameters

Each sensor pipeline in AuraRing transduces the incident magnetic field to a digital value. For a given ring pose, we need to be able to estimate the values reported by each sensor. Doing so accurately requires additional calibration parameters that define the *sensor model*, which describes how a given pose manifests as sensor measurements. These parameters model effects like the sensitivity of each coil and the relative positioning of the sensor coil on the PCB. Altogether, they capture offsets between the motion capture positions and the effective magnetic origins as well as effects of the AuraRing analog signal processing pipeline. Below, we briefly outline these parameters and how they fit into the AuraRing sensor model.

Coordinate System Offsets

Due to manufacturing tolerances, the exact position of each sensor with respect to origin reported by motion capture may not exactly align with the computed values from the CAD design. Consequently, we define offset positions and orientations to refine each sensor pose in the wrist frame. These slack terms are expected to be small — on the order of a couple of millimeters or degrees.

$$\vec{s}_w = \vec{s}_{w,CAD} + \vec{s}_{w,\mathit{offset}} \tag{4.6}$$

$$\mathbf{s}_w = \mathbf{s}_{w,\mathit{offset}} \times \mathbf{s}_{w,CAD} \tag{4.7}$$

We use a similar technique to define the dipole coordinate frame with respect to the ring coordinate frame.

$$\vec{d}_w = \vec{r}_w + \mathbf{r}_w \vec{d}_r \tag{4.8}$$

$$\mathbf{d}_w = \mathbf{d}_r \times \mathbf{r}_w \tag{4.9}$$

Modeling Analog Signal Chain

We then use the model in Section 4.2.2 to compute B_s, the magnetic field in the sensor reference frame. However, the sensors and amplitude demodulation pipeline do not represent a perfect measurement of the magnetic field. For one, the rectifier-based demodulation scheme only provides an unsigned estimate of the field strength along each axis. Moreover, each channel has a slightly different gain, due to different sensitivities and manufacturing tolerances of both the sensor coil and amplifiers. We also model the effects of noise and the diode forward voltage drop using a model that was validated for electromagnetic tracking in [142]. Specifically, we estimate the sensor measurement (x_i) as a function of the field ($B_{s,i}$) and the channel gain (g_i), noise (n_i), and bias (b_i).

$$x_i = \sqrt{(g_i \times B_{s,i})^2 + n_i^2} - b_i \tag{4.10}$$

Altogether, this model includes 23 parameters that must be learned for each sensor coil.

4.4.3 Learning Sensor Model Parameters

To learn these parameters, we collected a dataset that captures a wide range of motion of the ring. One of the authors wore the wristband while using the other hand to move and rotate the transmitter freely within in a volume around the hand. During this process, the ring and wrist pose are captured by the motion capture system and aligned as previously described. The data was explicitly collected without wearing the ring so that this dataset contains 6-DoF motion of the ring. Otherwise, the data would constrained by the 4-DoF wrist and finger kinematics of a particular user and it may be possible to overfit these parameters to that user.

We took a random sample of 30k frames from this dataset to use for learning and formulated the problem as a non-linear optimization problem, which we solve using Ceres Solver [4]. We learn the 23 parameters for each sensor coil independently. In doing so, the solver seeks to find the parameter set which minimizes the difference between the observed sensor values and the estimated sensor values, computed according to the observed pose and the sensor model. The solver uses the Levenberg-Marquardt trust region algorithm for minimization.

Upon convergence, this results in a set of parameters that can be used to estimate the sensor measurements as a function of pose. Figure 4.8 shows the correlation between the sensor estimates and actual measurements on this dataset before and after applying the sensor model presented here. Before applying the sensor model, the estimates have a 0.564 Spearman's rank-order correlation with the actual measurements. After applying the sensor model, the correlation increases to 0.995, validating the effectiveness of this model.

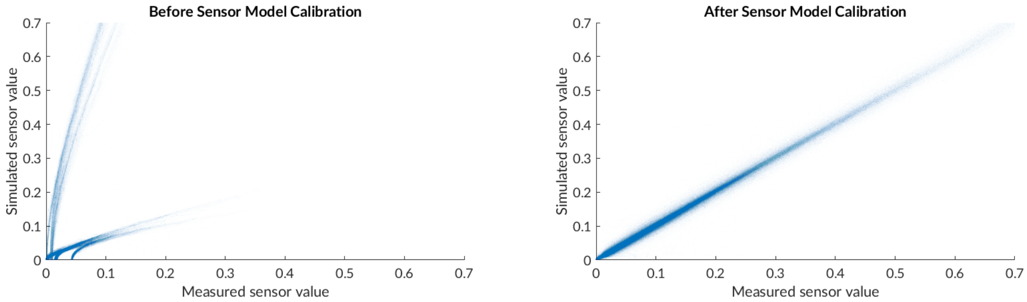


Figure 4.8: The one-time factory calibration process is necessary to match the synthetic data to the data measured by AuraRing.

4.5 Tracking Algorithm

Taken together, the magnetic field model and sensor model represent a forward model of the system—they estimate the sensor values given the ring pose. To track the ring, we must solve the reverse problem—estimating the ring pose given the sensor values. We implemented two different tracking algorithms to solve this reverse problem and estimate the 5-DoF pose of the ring. The first is an iterative optimization-based solution and the second is a neural network that approximates a closed-form solution to the reverse problem.

4.5.1 Approach 1: Iterative Model

The iterative model is similar to the method of learning calibration parameters discussed in Section 4.4, but operates on each frame separately. We use a non-linear optimizer [4] with the Levenberg-Marquardt algorithm to iteratively find the most likely pose. When a new set of sensor data arrives, the solver minimizes the error between the observed sensor values and the sensor values predicted by the forward model (Equation 4.10). It is allowed to run for a maximum of 100 iterations. For the first frame, the initial pose state is set to a default pose with the index finger pointing forward without bending. For subsequent frames, the solver uses the solution to the previous frame as the initial state. Further improvements in speed and accuracy may be possible by using a Kalman filter to proactively estimate the next ring pose. Because the sensor measurements in the presence of a weak field fundamentally have a lower signal-to-noise ratio (SNR), we do not want these measurements to significantly effect our measurements. Moreover, because Equation 4.10 has a minima when the field is null, there is the potential to get stuck in this minima. To reduce the impact of these effects, for any of the nine channels in which the measured magnetic field is extremely weak, we simply drop this term from the cost function.

To understand how the specific sensor calibration parameters affect tracking performance, we created a variant of the iterative tracking model that refines the calibration parameters based on two minutes of data from that session. This accounts for drift in parameters like the noise in the analog signal change, which can be impacted by environmental factors like temperature. We refer to the original session-independent iterative as ”pre-calibrated” and the session-dependent model as ”recalibrated”. Differences between the ”precalibrated” and ”recalibrated” models are likely due to manufacturing tolerances of this particular prototype. With additional engineering effort, we anticipate any performance gap here to be improved.

4.5.2 Approach 2: Neural Network-based Tracking

As an alternative to the iterative model, we used closed-form approximations to the reverse problem that are optimized for use on low-power devices. Specifically, we trained a neural network to predict both the 3 DoF position and 2 DoF orientation of the AuraRing ring directly from the magnetic measurements.

Generating Training Data

To eliminate any dependency on runtime training data, we train the network entirely from data generated by our simulated magnetic and sensor models. This gives us precise control over the sampling distribution of the training data and eliminates issues of over-fitting to a particular user's hand. We randomly generated different ring poses within a volume relative to the wrist that spans possible positions of the ring for most adults. Specifically, assuming the coordinate system defined in Figure 4.2, the generated data covers a volume defined by $-70mm < X < 90mm$, $-125mm < Y < 60mm$ and $-150mm < Z < -60mm$. We also add a random 3-DoF rotation to each point to complete the dataset of random poses.

We then use the sensor model to estimate the values of AuraRing's sensors at each pose. To improve performance, we further refine the dataset of poses to eliminate infeasible poses during normal operation by considering the direction of each field. Since AuraRing uses 3 three-axis receivers and each axis could be in or out of phase with the transmitter, there are $2^9 = 512$ possible combination of phase states for a given frame. However, due to the spatial geometry of our sensors and the kinematic structure of a hand, the majority of these states are not feasible during normal use. For example, because the ring always lies in front of (negative z-direction) the wristband, all of the z-channels of the sensors will have the same phase. In fact, after reviewing the generated data we found that only ten of these 512 combinations are feasible. We drop any data from the generated dataset that does not match one of these ten feasible phase states. Since training is done using simulated data, one could easily change these constraints or even add another state and train a larger network if the problem demands it. After this culling process, the dataset consists of 166,465 points.

NN Training

We propose to use two computationally simple models to regress to a position vector and the 2-DoF orientation of the ring. Both networks have a single hidden layer of 128 nodes to fit a function that maps the nine observed sensor values to a 3-dimensional position vector and to a 3-dimensions direction vector. We use a direction vector to specify orientation in order to remain invariant to the roll of the ring. Training using the synthetic dataset is performed in MATLAB with the Levenberg-Marquardt algorithm. With CPU training on a single core of a Xeon E3-1240 processor, training takes about six hours. Mean training error was $3.06\,\mathrm{mm}$ for the position model and $5.45°$ for the orientation model. Orientation error was calculated by computing the angle between the two direction vectors.

4.6 System Evaluation

4.6.1 Tracking Accuracy

User Evaluation Procedure

We evaluate the tracking accuracy of each of these models using data collected from users who we invited to try AuraRing. We do this to provide a realistic estimate of real-world performance under challenging conditions like different hand size, shape, flexibility, slippage of the devices on the hand and finger, and changes in environmental factors. We recruited 14 participants (5 M, 9 F) with different hand sizes to wear AuraRing while exercising the full range of motion of their wrist and fingers. The data collection was carried out in the same 10-camera motion capture lab used to calibrate the sensor model. The researchers helped the participants put on the wristband and placed the ring on the index finger of the right hand. To ensure no issues related to battery life, the ring was powered with a slightly

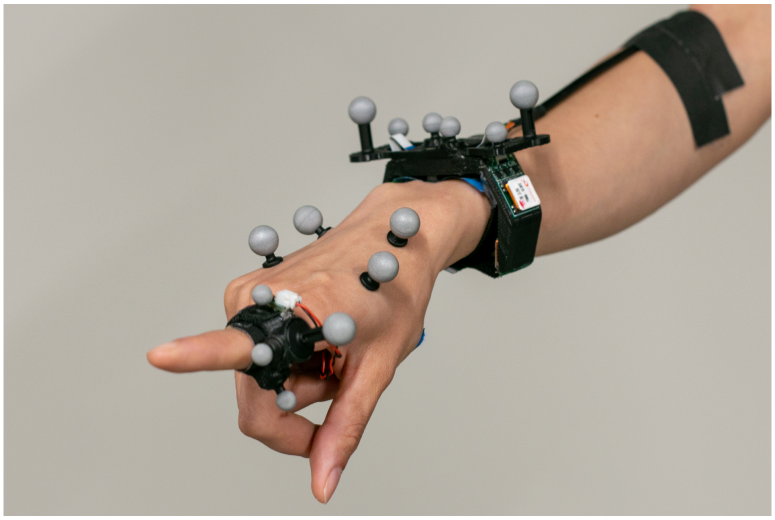


Figure 4.9: Data collection setup. IR retroreflective markers are placed on the ring, palm, and wristband to facilitate tracking. The user moves the index finger and wrist while measurements are collected. AuraRing streams data to the PC over a USB connection.

larger $105\,\mathrm{mA\,h}$ LiPo battery that they held within their hand. As before, markers were placed on an attachment to the ring and wristband to facilitate optical tracking. As showed in Figure 4.9, we placed additional four markers on the top side of the hand for debugging and visualization purposes. The participants were asked to freely and naturally move their wrist and finger for 10 minutes, while being sure to exercise all possible joint motion. During this time, the ground-truth pose and the magnetic sensors data was recorded by a Python program.

After processing the data, we dropped data from two participants due to poor quality of the optical tracking due to occlusions. Altogether this resulted in more than approximately two hours of data consisting of over 1.7 million synchronized data points.

Table 4.1: Shows the position and orientation error among 12 participants using iterative and neural network models

	NN Model	Precalibrated Iterative Model	Recalibrated Iterative Model
Mean X Error (mm)	2.85	1.65	0.50
Mean Y Error (mm)	4.18	3.41	1.21
Mean Z Error (mm)	1.80	1.14	0.51
Mean Position Error (mm)	6.07	4.41	1.53
Mean Orientation Error (degrees)	8.35	4.65	1.69

Results

We compare the tracking accuracy for both the iterative and neural network models as well as the session-dependent recalibrated iterative model. Table 4.1 shows the mean accuracy for the pose of the finger using each algorithm. The full error distribution is shown in the CDF in Figure 4.10 that aggregates data across all twelve participants. These results show a mean tracking error of $4.41\,\mathrm{mm}$ for the precalibrated iterative model and $6.07\,\mathrm{mm}$ for the neural network model. The error drops to $1.53\,\mathrm{mm}$ after fine-tuning the sensor model for each session.

For orientation, the iterative model tracks the forward direction of the finger with a mean error of 4.65° and the neural network tracks with a mean error of 8.35°. The orientation error drops to 1.69° with the recalibrated model. The full distribution of orientation error for each model is shown in Figure 4.11.

While accuracy is an important measure of tracking performance, the ability to track relative motion is essential if using AuraRing as an input device. Figure 4.12 shows the ground truth and estimated position over a representative few seconds of two represen-

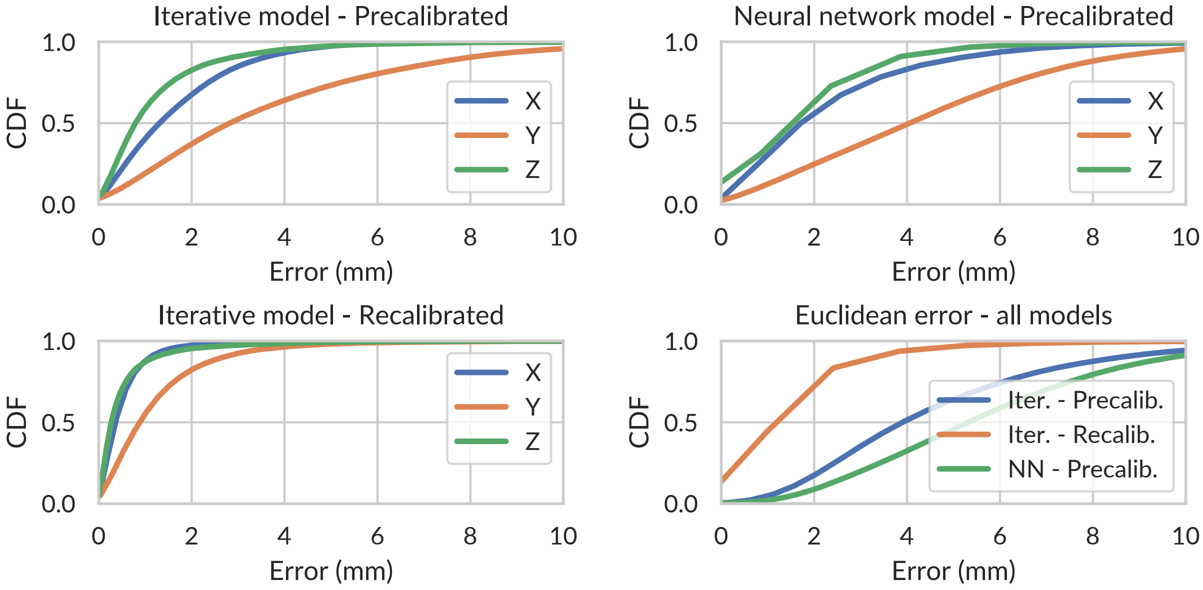


Figure 4.10: CDF of 3D position tracking among all participants using the iterative model vs the neural network model. The best accuracy is achieved by using the recalibrated iterative model

tative sample traces. The sample trace on the left, has a below average mean error on the iterative model of $2.56\,\mathrm{mm}$. The trace on the right performs worse at $4.99\,\mathrm{mm}$. Note that even in cases where the error is relatively high, the relative motion still tracks the ground truth motion. These traces do not have any additional Kalman filtering applied to them.

To better understand the dynamics of each model, Figure 4.13 shows a highly zoomed in trace of the x-direction only for a representative $200\,\mathrm{ms}$ window. The dark lines indicate estimates that have been smoothed with a Kalman filter. The lighter lines represent the raw estimates from each model. The estimated position is illustrated after implementing a Kalman filter which smooths out the estimated position. This shows that even in the case of a static offset in tracking, the relative pose tracks well, even over timescales of tens of milliseconds. This pattern holds for the neural network model as well.

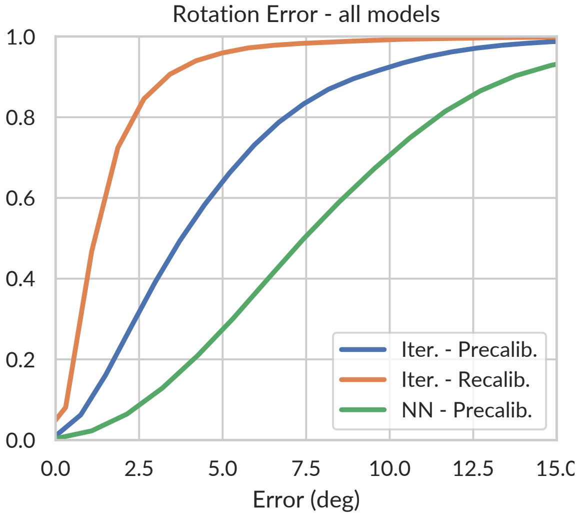


Figure 4.11: CDF of orientation tracking among all participants using all models

4.6.2 Resolution

Resolution is an important characterization for a sensor system that quantifies the smallest detectable change in the measure of interest. The use of high-precision electromagnetic sensors represents a significant advantages for AuraRing over other wearable sensing systems because of its ability to capture small and subtle interactions.

We used a motorized linear stage[2] to quantify the resolution of the AuraRing hardware. The stage is software-controllable and has a repeatability of $10\,\mu\mathrm{m}$ and a step size of $0.2\,\mu\mathrm{m}$. We affix the ring to the stage and place the wristband $125\,\mathrm{mm}$ away, a typical operating distance during normal use. The devices were oriented such that most of the magnetic flux is oriented in the z-direction of the wristband. We programmed the stage to move back and forth (in the z-direction) and variable step sizes ranging from $100\,\mu\mathrm{m}$ to $1\,\mathrm{mm}$. Figure 4.14 shows the values measured in one of the sensor coils during this procedure. No filtering was applied to this data. From this figure, one can observe that a step size of $1\,\mathrm{mm}$ is clearly visible. As the step size decreases to $100\,\mu\mathrm{m}$, the different in sensor reading approaches the

---

[2] PI VT-80

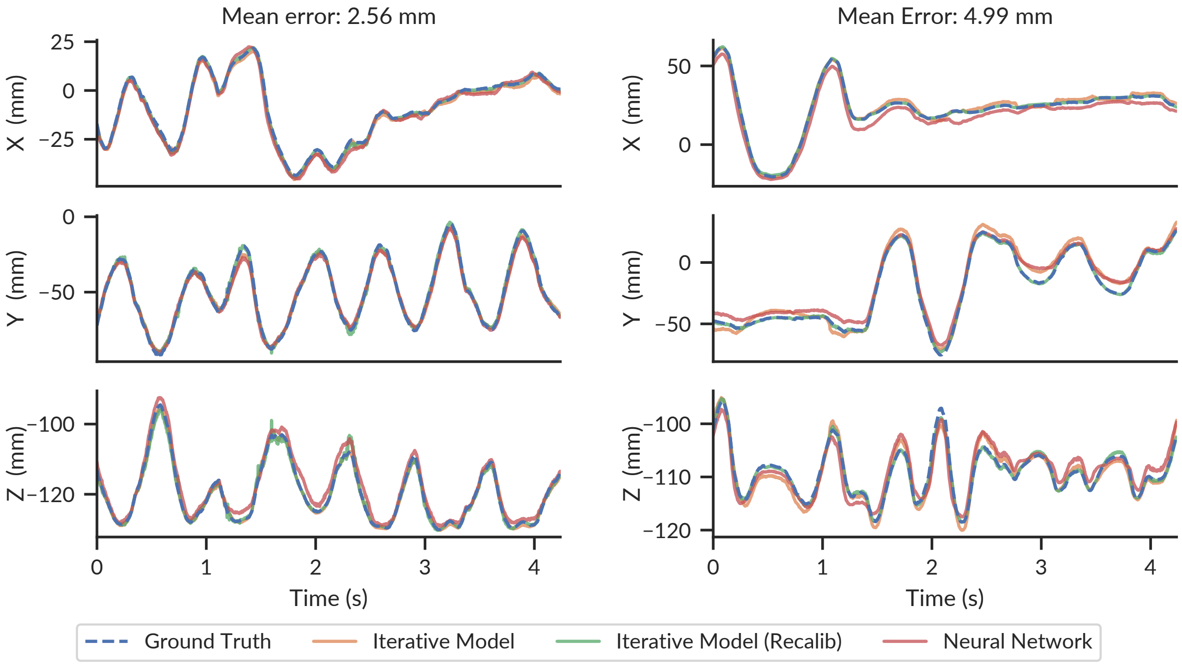


Figure 4.12: A representative few seconds of 3D positional tracking for two of the participants. While the tracking error is relatively higher for the one on the right, the relative motion still tracks the ground truth motion.

size of a single bit of the ADC.

We note that because the device is capable of sampling at speeds much faster than a typical user interface would require, the resolution could likely be significantly improved through filtering. We also note that because we do not have a ground truth measure of the AC magnetic field strength, the resolution here depends on the specific position at which it is measured. Nonetheless, this is a representative example that illustrates the sensitivity of the hardware platform.

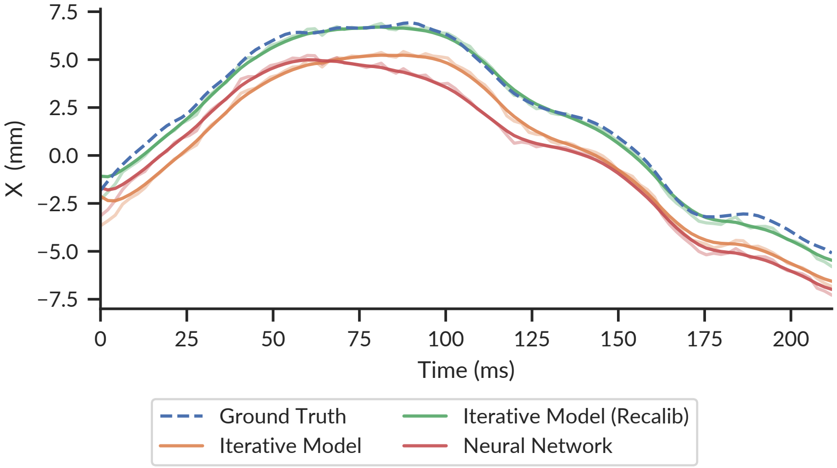


Figure 4.13: A zoomed in trace for the x-direction. AuraRing leverages a Kalman filter to smooth out the estimated position. Even when the neural network has a static offset, the relative motion is preserved.

4.6.3 Power

We measured the power consumption of both the ring and wristband of AuraRing using a USB oscilloscope to measure the voltage drop across a series resistor of values $9.8\,\Omega$ and $109.5\,\Omega$ for the ring and wristband respectively. The ring consumes only $715\,\mu\mathrm{A}$ of current ($2.34\,\mathrm{mW}$) and the wristband consumes $22.2\,\mathrm{mA}$ ($73.3\,\mathrm{mW}$). Two $12\,\mathrm{mA\,h}$ batteries that fit comfortable on the ring PCB allow the ring to operate continuously for about 17 hours. A $2\,\mathrm{W\,h}$ battery would allow the wristband to continuously operate for more than a full day. In practice, such a system could have a sleep functionality so that the transmitter and processing circuitry do not need to be on when not in use.

4.6.4 Speed and Compute

Although the analysis presented in this work was performed offline, the algorithms used for pose estimation were designed with real-time operation in mind. The iterative approach

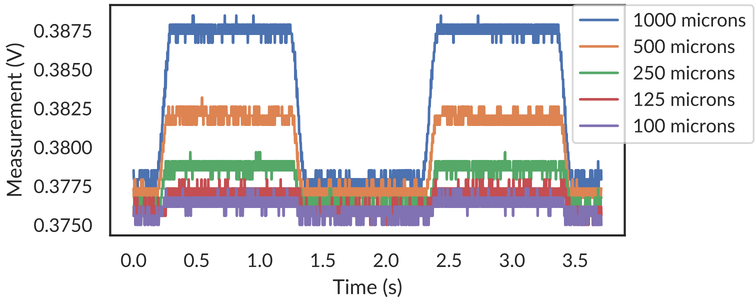


Figure 4.14: The AuraRing hardware is sensitive to 100 microns of movement. A single channel of one of the sensors depicted as the transmitter is moved by a motorized linear stage.

performs the best, but requires the most compute power. However, despite the iterative nature of the algorithm, it easily exceeds realtime speeds. When fed each data frame sequentially, the algorithm operates at speeds exceeding $1\,\mathrm{kHz}$ on a single core of a Xeon E3-1240 processor. We anticipate that, with some care, it is possible to port this algorithm to a mobile processor.

However, in scenarios where compute is at a premium, such as on a mixed-reality HMD, we designed the neural network approach for maximum efficiency. The network contains only a single layer with 128 hidden nodes. Runtime position and orientation estimation consists of just two matrix multiplications for each (9×128 and 128×3).

4.6.5 Interference

Like all electromagnetic tracking systems, AuraRing's performance will degrade in the close proximity to metallic, particularly ferromagnetic, objects. Unlike outside-in solutions like Polhemus [96], the AuraRing transmitter and receiver are in relative close proximity, which significantly reduces the scope of possible interference sources.

To quantify the effects of large metallic objects on AuraRing, we measured the effect of two commonly used devices i.e a smartphone and a laptop that are likely to be in close

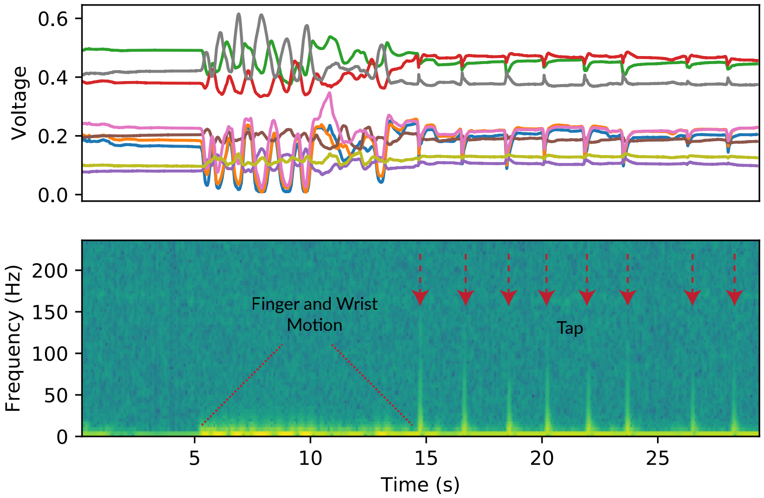


Figure 4.15: AuraRing is capable of measuring high-speed events such as taps. A user performs multiple taps after moving their finger in the air. (Top): nine raw sensor measurements. (Bottom): Spectrogram of the recorded data.

proximity to the AuraRingduring operation. The Aura transmitter and receiver are placed on a flat space. One of the researchers brought the devices close to the wristband and ring one at a time. We observed changes in the received signal at a distance of about $15\,\mathrm{cm}$ for both devices.

4.7 Additional Functionality

Having the ability to track one's finger precisely without line-of-sight enables a wide range of applications. However, we believe that the speed and precision which AuraRing's magnetic tracking enables unlocks additional functionality not commonly found in wearable sensing systems. In this section we discuss and demonstrate a few additional features that AuraRing can enable.

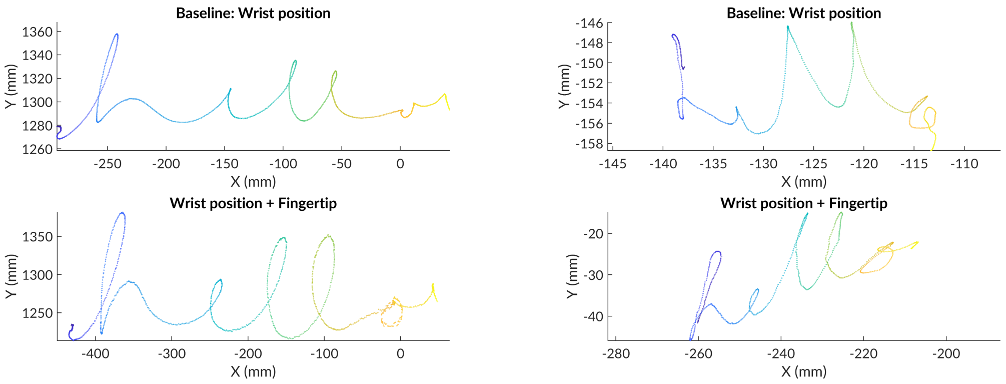
4.7.1 *Tap Detection*

Visions of free-hand interaction techniques in AR and VR commonly include in-air taps and gestures. However whether you are clicking in an app on your AR glasses or drawing in a VR game, the sensation of touch is one of the critical factors for an immersive experience. The ability to robustly detect taps on environmental and body surfaces enables new kinds of always-available ambient interfaces [45, 149]. AuraRing helps enable this future by providing not only an estimate of finger position, but a high-speed robust signal which can be used to detect taps.

With a data rate of $472\,\mathrm{Hz}$ and a sensor bandwidth much higher than this, AuraRing is capable of measuring high speed events. To demonstrate this, Figure 4.15 shows a series of taps of varying intensities on a tabletop surface after a period of freely moving the finger and wrist in space. The lower portion of the figure shows a spectrogram with a window size of 128. While hand and finger motion contain mostly low frequency content, taps are immediately distinguishable due to their broadband spectrum and appear to contain content up to about $150\,\mathrm{Hz}$.

4.7.2 *Free-form and Subtle Hand Writing*

In today's AR/VR systems, hand writing is usually done by tracking the wrist using cameras on the HMD. Although this method would work for dexterous and big wrist motions, it fails to track subtle handwriting done with minimal wrist motion.


Baseline: Wrist position

Wrist position + Fingertip

Baseline: Wrist position

Wrist position + Fingertip

Figure 4.16: Handwriting examples using wrist-only motion frop motion capture (top) and wrist motion + AuraRing (bottom). The left example shows large hand motion. The right example show small writing on a handheld object.

To demonstrate that AuraRing is useful for much more than just gesture recognition, we show a few example traces of handwriting reconstruction using AuraRing. We consider two types of use cases. For use with a mixed reality HMD, we anticipate the use of wrist-tracking using fiducial markers embedded in the wristband. While one could use the wrist position as a virtual ”pen”, we argue that the dynamics of the wrist and finger joints are crucial to enabling a compelling experience. In Figure 4.16, we show two examples of writing the word ”hello” in the air; one with big large movement over tens of centimeters (left) and another that was ”written” on a notebook held in the opposing hand spanning just a couple of centimeters (right). The top row shows the result if one were to use just the wrist position; in this case, the wrist pose was captured from the motion capture system. On large motions, wrist position is a reasonable proxy that might be useful for gesture recognition, but not for precise input. For smaller motions the wrist position results in a poor quality signal. However, as shown in the bottom row, combining the estimate of finger position from

In other use cases, the wrist may not be tracked, such as with interaction on a smartwatch. In these cases, AuraRing is still capable of providing an accurate reconstruction of relative fingertip position, which is still useful for handwriting tasks. Figure 4.1 (left) shows an example of a handwriting trace reconstructed without any optical motion capture system. AuraRing results in a significantly more useful signal. With appropriate visual feedback, we anticipate the performance to be even better.

4.8 Conclusion

In this work, we present AuraRing, a wearable ring and wristband that enables precise, subtle and accurate finger tracking. Using minimal, low-power electronics on the ring, AuraRing can operate for about a day on self-contained batteries. We presented two methods for tracking, a neural network-based approach and an iterative model. AuraRing only needs a one-time factory calibration process to match its synthetic data to sensor measurements and can track with a mean error of $4.4\,\mathrm{mm}$. For additional performance, recalibration can be used to achieve a mean positional error of $1.5\,\mathrm{mm}$.

Chapter 5

CONTINUOUS INFRARED WRIST ANGLE TRACKING USING A WRISTBAND

5.1 Introduction

Wearable devices have grown immensely popular and are enabling computing in many areas of people's lives across a wide range of scenarios. Tracking the movement of the hand has long been important in developing natural and intuitive interaction paradigms for computing [108]. Specifically, the dexterity of human wrist joints typically enables a broad range of motion [74], making the wrist a promising modality to drive input for wearable computing.

Existing hand tracking methods most often rely on computer vision [99, 21, 135, 127, 79, 101] with cameras in the environment or on a head-mounted display (HMD). These approaches yield high accuracy, but require line-of-sight and, in some cases, infrastructure support, that limit mobility and versatility. In addition, many commercial AR/VR systems primarily use a handheld controller to continuously track users' hand pose. While these solutions offer great accuracy and specific affordances, they are still limited by mobility constraints. Tracking the hand without controllers and without line-of-sight affords the potential to consider new freehand interactions. The growing demand for computing in ubiquitous contexts motivates researchers to consider wrist-worn devices (e.g., smartwatches and smart bands), which have become increasingly mainstream. However, most research efforts around wristband-based sensing focus on discrete, gesture based interactions [159, 50, 129, 26, 51, 152, 36, 35]. Although discrete gestures are useful, they are just one part of the interaction language needed for mobile wearable computing. For common tasks like pointing at targets, drawing, sliding, or swipe-based text input, an input device that supports continuous tracking will be most appropriate.

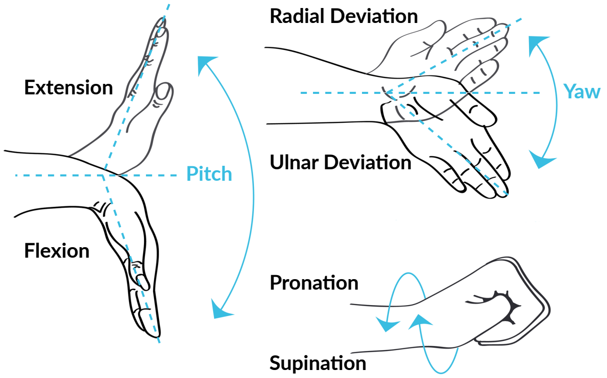


Figure 5.1: Diagram illustrating different wrist motions. RotoWrist continuously captures wrist flexion/extension and radial/ulnar deviation. It does not attempt to capture pronation/supination.

In this work, we present RotoWrist[105], an infrared (IR) light-based device that performs continuous 2-degree-of-freedom (DoF) wrist tracking from the wristband. RotoWrist works without requiring any per-user or per-session training. The RotoWrist system consists of a wristband that uses eight ToF IR modules to continuously estimate the 2-DoF angle of the wrist with respect to the forearm in real-time. Each low-power sensor measures the absolute distance from the hand. As the user moves their wrist, the relative distance between the hand and each sensor changes. RotoWrist combines these eight measurements into a 2-DoF wrist orientation—flexion/extension and radial/ulnar deviation, as depicted in Figure 5.1[1].

Among related work in sensor-based tracking of the wrist, RotoWrist is most similar to WristWhirl[36]. In RotoWrist, we prioritized robustness and accuracy and designed the system from ground up to support continuous, absolute tracking without any user calibration. While the WristWhirl system is capable of sensing continuous wrist motion, it did

---

[1] Image has been copied from https://hackmotion.com/role-of-wrist-angles-in-golf-swing/

not investigate tracking performance and focused primarily on the accuracy of detecting 8 distinct gestures. Not only does RotoWrist support real-time continuous tracking that is benchmarked against a ground truth system, but we further establish RotoWrist's ability to operate in a cross-session and cross-user fashion. Finally, we demonstrate that RotoWrist can be used for pointing tasks.

RotoWrist offers a rich input source for a variety of wearable devices, including smartwatches and HMDs. With RotoWrist, a user can provide continuous input using their wrist, in eyes-away contexts where line-of-sight to a camera might be difficult to maintain. For example, user might point while keeping their arm motionless at their side or with their hands beneath a table.

Our primary contributions are:

1. A simple, low-power hardware architecture consisting of eight time-of-flight IR light modules embedded in a custom-built wristband that enables continuous wrist angle tracking.

2. A tracking algorithm that can reliably estimate 2-DoF orientation of the wrist without per-user or per-session training.

3. A system characterization and user evaluation demonstrating tracking accuracy of 5.9° in flexion/extension and 6.8° in radial/ulnar deviation across users without any calibration, compared to an optical motion capture system.

4. A study comparing RotoWrist's pointing performance to a high-precision wrist and forearm tracker based on a motion capture system.

5.2 Implementation

We use a ToF-sensing array to track wrist orientation by comparing the distance measurement from each sensor to the hand. The conical fields of view of each individual time-of-flight sensor overlap significantly, so several sensors simultaneously produce an in-range

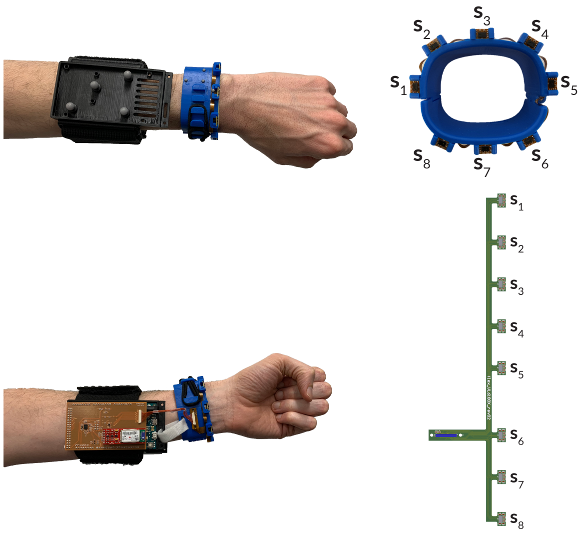


Figure 5.2: RotoWrist consists of a controller arm band that handles communicating the data back to host PC and eight time-of-flight modules embedded in a wristband.

sensor measurement. As parts of the hand move in and out of the field of view of each ToF sensor, the distance reading from that sensor becomes smaller and larger. The RotoWrist system consists of a sensing wristband that incorporates eight ToF IR modules and a controller arm band that handles powering the sensing wristband and communicating the data back to a host PC. The following sections provide details of the RotoWrist hardware, capabilities and algorithm.

5.2.1 *Sensing Wristband*

One of the challenging aspects of designing a wristband is to manufacture one that can fit different people and whose sensing is robust to different wrist sizes. In addition, the form-factor of the wristband should be minimally invasive. We design the wristband as a top and bottom part that connect together using a hinge mechanism. All parts are made using a 3D printed Nylon material. The hinge (Figure 5.2 right) on the side allows the band to open and close. On the other side, we have leveraged an elastic band to securely fasten the top and bottom parts. As shown in Figure 5.2 left, there are several hooks on the top part that the elastic band can be connected to based on the wrist size. To further support people with different wrist sizes, we built two sizes of the sensing wristband – one with a diameter of $120\,\mathrm{mm}$ and the other $180\,\mathrm{mm}$.

As shown in Figure 5.2 (bottom right), we built a custom flex PCB that incorporates eight time of flight modules (VL6180X) and a 22-pin SMD connector. These modules are equally spaced around the wristband and they sit perpendicular to and at most $8\,\mathrm{mm}$ from the wristband's surface. Each individual sensor determines the distance to the nearest object within a $25°$ cone by emitting pulsed infrared light and timing the returned reflection. The eight TOF modules consume only $13.6\,\mathrm{mA}$.

5.2.2 Controller Armband

The sensors on the wristband are connected to a custom made daughterboard through a very thin 22 pin assembly cable (A22XSR22XSR36R254B). Since the recommended operating voltage for the ST ToF moudel is $2.8\,\mathrm{V}$, the daughterboard also incorporates two bidirectional level-shifting translators (TXB0106). As shown in Figure 5.2 (bottom left), the daughter board sits on an Arduino Due. The sensors on the wristband are connected to the microcontroller (Atmel SAM3X8E ARM Cortex) on the Due board over $\mathrm{I^2C}$, which triggers measurement. A timer-based algorithm is set to take measurement every $20\,\mathrm{ms}$ from all the sensors resulting in a tracking frame rate of $50\,\mathrm{Hz}$. A SparkFun Bluetooth Mate Silver module also sits on top of the daughter board to facilitate sending the data to a host PC over Bluetooth. Using Velcro tape, the armband also hold a small battery pack that powers the Arduino and every other component on the other boards.

5.2.3 Tracking Algorithm

We use a simple tracking algorithm that reliably tracks wrist orientations across different users. As shown in Figure 5.2, we name each sensor in a clockwise fashion starting on the radial side from S_1 to S_8. As expected, we observe that the topmost (S_2, S_3, S_4) and bottommost (S_6, S_7, S_8) sensors have the most linear effect when there is a flexion/extension activity (i.e. pitch) and the sensors on the radial (S_8, S_1, S_2) and ulnar (S_4, S_5, S_6) sides are most responsive to radial/ulnar deviation (i.e. yaw). We model these observations using Equations 5.1 and 5.2 where D_{S_i} refers to the distance value that is returned by the sensor S_i at a certain point in time. For each of the equations, there is a weight term associated with each of the sensors ($W^p_{s_i}$ and $W^y_{s_i}$) which defines the individual weight given to a particular sensor and (B_p and B_y) refer to a bias offset value.

$$Pitch = \left(\sum_{i=2,3,4} \frac{W^p_{s_i}}{D_{S_i}} - \sum_{i=6,7,8} \frac{W^p_{s_i}}{D_{S_i}} \right) + B_p \tag{5.1}$$

$$Yaw = \left(\sum_{i=1,2,8} \frac{W^y_{s_i}}{D_{S_i}} - \sum_{i=4,5,6} \frac{W^y_{s_i}}{D_{S_i}} \right) + B_y \tag{5.2}$$

These 14 parameters (7 each in pitch and yaw, 6 individual sensor weights, and 1 bias term) are learned in a user evaluation explained in Section 5.4. We formulated the problem as a non-linear optimization problem which we solved using a Basin-hopping algorithm.

5.3 User Evaluation Setup

A ground truth system is required to evaluate RotoWrist orientation accuracy. Since the wrist is not a rigid body and the skin is stretchable, tracking the angle of the wrist with respect to the arm is challenging. The most precise method of measuring the wrist angle is to use radiographs to measure the carpal bone angle on lateral wrist [65]. Prior research has looked at building a custom-made calibration device that has pivot joints with potentiometers to measure the wrist angle [132]. Since building such devices would affect the

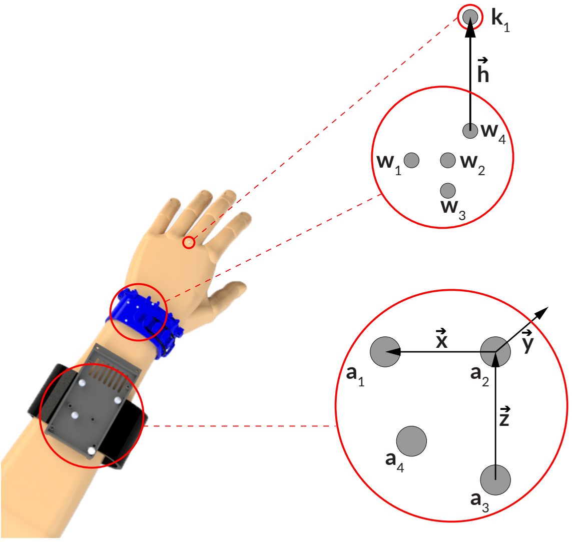


Figure 5.3: Retro-reflective markers placed on the wristband, armband and middle finger knuckle to facilitate tracking.

IR reading from our wristband (due to the calibration device occluding the IR sensors) and doing radiographs is not practical, we used a fifteen-camera OptiTrack[2] motion capture system to record the real-time pose of the wrist, forearm, and hand stub at $240\,\mathrm{Hz}$. We placed retro-reflective spheres on both the controller armband and sensing wristband in known locations, as shown in Figure 5.3. The participant wore the wristband and armband on their left hand and a researcher placed a small IR retro-reflective marker on their middle finger metacarpophalangeal joint.

We placed the armband retro-reflective markers in a way that two of the four are in the direction of the arm (markers a_2 and a_3), and two are perpendicular to the arm (markers a_1 and a_2) as shown in Figure 5.3. The markers ($a1, a2, a3$) define our arm coordinate frame:

$$\vec{z} = a_2 - a_3 \tag{5.3}$$

---

[2] OptiTrack Prime 17W

$$\vec{x} = a_1 - a_2 \tag{5.4}$$

$$\vec{y} = \vec{z} \times \vec{x} \tag{5.5}$$

We also placed four retro-reflective markers on the sensing wristband. Markers w_1, w_2, and w_3 are used to help track the arm orientation in Section 5.5. w_4 is placed on the top middle part of the wristband and, together with k_1, defines the wrist orientation vector h:

$$\vec{h} = k_1 - w_4 \tag{5.6}$$

The placement of w_4 and k_1 markers is chosen so that when a person has a neutral hand pose (i.e. zero pitch and yaw), $\vec{h}$ is parallel to $\vec{z}$ and perpendicular to $\vec{x}$. Given the wrist orientation vector $\vec{h}$ and the arm coordinates ($\vec{x}, \vec{y}, \vec{z}$), pitch and yaw of the wrist can be defined as follow:

$$pitch = \arccos(\vec{h}.\vec{y}) \tag{5.7}$$

$$yaw = \arccos(\vec{h}.\vec{x}) \tag{5.8}$$

5.4 User Evaluation 1: Angular Accuracy

We performed two focused studies to evaluate the performance of RotoWrist. The first study evaluates the dynamic tracking accuracy of our system compared to a motion capture ground truth system. Maintaining performance across worn sessions. (i.e., reworn at a later time) or between different people is a challenge for almost all bio-sensing and wearable systems, since misalignment could result in signal change. The following sections evaluates the robustness of RotoWrist in different scenarios.

We recruited 14 people (25-59 years old) to evaluate the tracking accuracy of our system. The study took less than 30 minutes and participants were compensated for their time. Based on their wrist sizes, five people wore the smaller sized wristband and the other nine were given the larger wristband. Upon arrival, we asked the participants to sit on a chair while putting their left arm on a table that was in front of them. A researcher helped the participants put on the wrist and arm band and the retro-reflective marker on the left hand of the user as explained in Section 5.3. The wristband was placed where a user might typically wear a smartwatch (roughly at the tip of the ulna) as depicted in Figure 5.2. Using the elastic bands, the wristband was fit to the user making sure they can move their wrist without discomfort. The location of the armband is roughly placed in the middle of the forearm in a way that the markers are facing the user. For consistency, we asked users to maintain a fist pose throughout the study. Anecdotally, RotoWrist is robust to different finger poses, though we have not fully evaluated this effect.

The evaluation was conducted in two sessions and each session had two phases. *Phase 1, Defined Motion*: Participants were asked to hold their wrist in a neutral position in the beginning of each session and then perform three flexion/extension and three radial/ulnar deviation movements, exercising their full, comfortable range of motion. *Phase 2, Random Motion*: Participants performed 4 minutes of free-form movements while being asked to exercise all possible wrist joints motions. After the two phases in the first session, the wristband was taken completely off. The users were instructed to take a minimum of 2 minutes of break. After the break, the researcher asked the participants to put the wristband back on, and the same procedure was given to the user and the two phases were repeated.

To time-synchronize the sensor stream from RotoWrist ($50\,\mathrm{Hz}$) with the ground truth wrist orientation stream ($240\,\mathrm{Hz}$) given by Equations 5.1 and 5.2, we introduced a time offset between that maximizes correlation between the orientation measured from motion capture system and the orientation estimated through Equation 5.9 and Equation 5.10.

$$Pitch_{apx} = 150\left(\sum_{i=2,3,4}\frac{1}{D_{S_i}} - \sum_{i=6,7,8}\frac{1}{D_{S_i}}\right) \tag{5.9}$$

$$Yaw_{apx} = 200\left(\sum_{i=1,2,8}\frac{1}{D_{S_i}} - \sum_{i=4,5,6}\frac{1}{D_{S_i}}\right) \tag{5.10}$$

After synchronization, we formulated the problem as a non-linear optimization problem which we solved using a Basin-hopping algorithm. The cost function for this optimization problem was set to be the difference in the pitch and yaw estimation from Equation 5.1 and 5.2 and the angles calculated from motion capture system (Section 5.3).

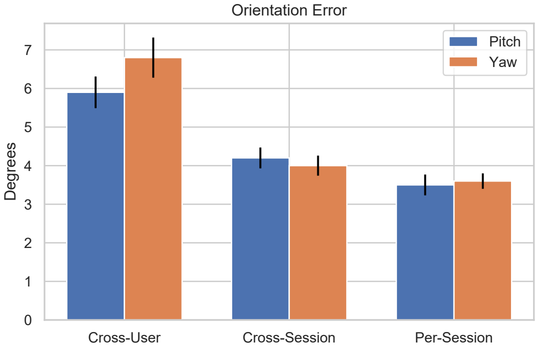
Although using a motion capture system to capture the ground truth of the orientation of the wrist with respect to the forearm is practical, this method does not capture the full complexity of the human hand anatomy. For example, in a neutral pose (i.e both pitch and yaw of the wrist are zero) we expect that the $\vec{h}$ is perfectly aligned with $\vec{z}$. This is not always true due to the geometry of the hand. Therefore in evaluating RotoWrist, we add a one time offset, to force the pitch and yaw estimation from Equations 5.1 and 5.2 to start at zero. This offset is only required to compare RotoWrist estimates with the wrist orientation derived from ground truth and is not necessary for normal operation. We evaluate the tracking accuracy of RotoWrist in three different categories: 1) cross-user where no calibration is needed, 2) cross-session, and 3) per-session where a user can perform a short calibration.

5.4.1 *Cross-User Evaluation*

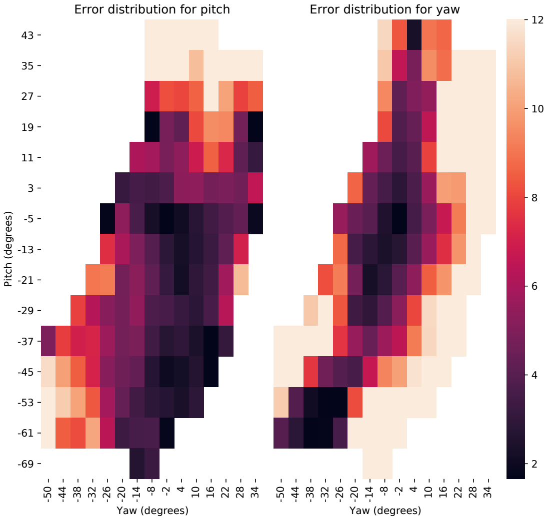
As stated in Section 1, a widespread input solution should be robust and extensible (e.g. supporting different users and contexts out of the box). Therefore, in this section, we evaluate how our tracking holds between users. Notably, we learned the heuristic model parameters presented in Section 5.2.3 on only Phase 1 (Defined motion) from the first session of all participants except one and tested on all of the data (including phase 1 and phase 2) from the left out user. We repeat this for all of the participants. We learn a total of 14 parameters for this evaluation: 12 individual sensor weights, and 2 bias terms. The measured RMSE for pitch and yaw across users is 5.9° and 6.8° respectively.

5.4.2 *Cross-Session Evaluation*

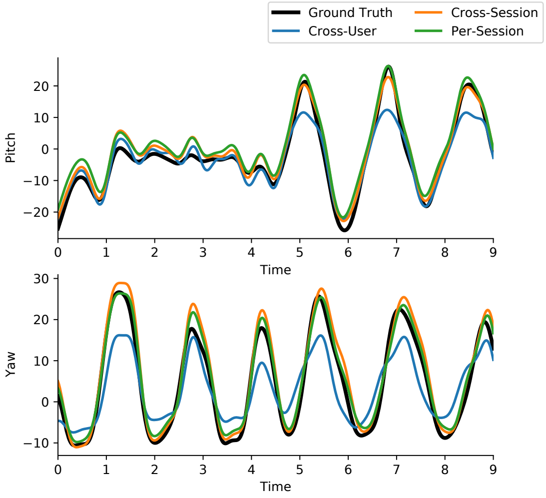
To quantify how the performance holds when reworn by the same user, we ran a leave-one-session-out cross validation for each of our participants. We learned the same heuristic



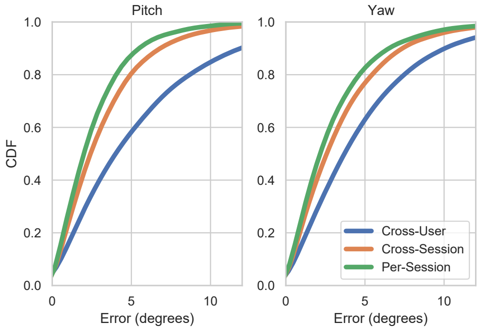
(a) Shows the orientation error and standard deviation among 14 participants using the three models.



(b) Heatmap showing spatial distribution of error among all users for cross-user evaluation



(c) A representative few seconds of 2D orientation tracking for one of the participants (mean error for cross-user: 5.7°).



(d) Cumulative distribution function of pitch and yaw orientation tracking error.

Figure 5.4: Angular accuracy evaluation results

model parameters presented in the previous Section 5.4.1 on only Phase 1 (Defined Motion) from session one. After learning these parameters, we test on all data (Phase 1 and Phase 2) from session two. The pitch and yaw root mean square error (RMSE) in degrees for cross-session accuracy across all users is 4.2° and 4.0° respectively.

5.4.3 Per-Session Evaluation

Although we don't expect users to calibrate RotoWrist each session, to better shine a light on how the tracking improves with a very small calibration procedure, we evaluate RotoWrist's per-session accuracy. To do this, we learned the heuristic model parameters presented in previous section from Phase 1 of each session and tested on Phase 2 of the same session. The pitch and yaw RMSE in degrees for per-session using this calibration reduces to 3.5° and 3.6° respectively. We summarized these results in Figure 5.4a.

A representative few seconds of orientation tracking for one of the participants is shown in Figure 5.4c. Even in the most challenging condition with no calibration, relative motion are preserved. This will help with pointing tasks where the user can see where the pointer is.

Figure 5.4d shows the cumulative distribution function (CDF) of orientation accuracy between all three evaluations discussed above. Figure 5.4b shows the spatial distribution of orientation tracking error of the cross-user model projected onto a 2D plane among all users. As depicted and expected, due to kinematics of the wrist, users performed more negative yaw (flexion) and negative pitch (radial deviation) movement which is aligned with the functional ranges of the wrist joint motions [104]. Due to the geometry of the wrist, the trapezoid bone and the thumb are placed slightly outside of the forearm axis and because of that the performance of RotoWrist is relatively better in radial than ulnar deviation as shown in this Figure 5.4b.

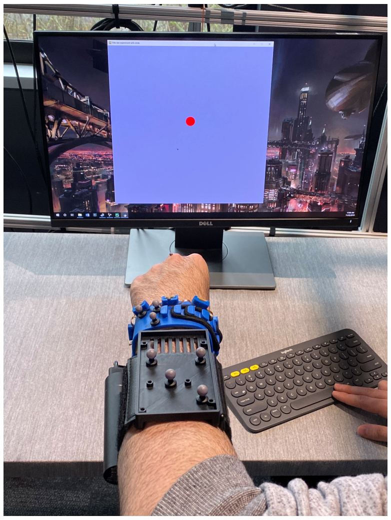


Figure 5.5: 2D pointing study setup. Users sit on a chair while performing the trials and use the space bar of a keyboard for target selection.

5.5 User Evaluation 2: Pointing

We conducted a second study to evaluate RotoWrist on a pointing task. Through this study we aim to understand whether RotoWrist's accuracy is sufficient for a pointing task and how this performance compares with respect to ground truth wrist tracking and ground truth forearm tracking. Forearm tracking can also be enabled via wrist-worn sensing [113, 3], but involves large, fatiguing arm movements, and thus is a contrasting point of comparison to wrist tracking. We further evaluate how well RotoWrist pointing adheres to Fitts' law.

5.5.1 Design

Since the yaw motion range is smaller than the pitch, more research is needed into how the angular movement across different wrist rotational axes is mapped to the on-screen cursor. Prior work [40] has shown that a mismatched mapping function between the different wrist motion axes can still yield useful results. To keep things simple, we conducted the pointing evaluation along the yaw and pitch axes separately. For each of the two axes (yaw and pitch), we had three independent variables: *Method* (GTF - Ground Truth Forearm Tracking, GTW – Ground truth Wrist Tracking, R - RotoWrist), *Distance* (Pitch: 100, 200, 400 pixels; Yaw: 50, 100, 200 pixels) , *Width* (20, 40, 60 pixels) resulting in $2 \times 3 \times 3 \times 3 = 54$ conditions. The distance and width conditions allowed us to investigate a realistic range of task difficulties. The Yaw and Pitch distances are different since yaw has a lower angular range of wrist motion. Based on initial pilots, we mapped the wrist and forearm motion to pixels as follows: a single degree of angular wrist motion along the yaw-pitch axis translated to 4.35 pixels along the x-y axis. For forearm tracking, a 1cm arm motion along the horizontal-vertical axis translated to 20 pixels along x-y.

For each condition, participants performed 5 repetitions. We followed a within-subjects design where each participant performed all conditions. We counterbalanced Axis and Method among the participants using a Latin square and randomized the Distance and Width. We recruited 12 participants (25-50 years of old) for the study. All except one was right-handed. The study took 75 mins and participants were compensated for their time. In total, we had 12 participants $\times$ 54 conditions $\times$ 5 repetitions = 3240 trials.

5.5.2 *Procedure*

Participants sat on a chair in front of a computer display. To allow participants to use the keyboard on a hard surface, we placed a table in front of them as shown in Figure 5.5. We used the same data collection setup explained in Section 5.3 and calculated ground truth wrist orientations using Equations 5.7 and 5.8. The study software was written in PyGame.

Prior to beginning, participants were given a short instructional session in which they familiarized themselves with the tasks and how should they move their wrist or arm to move the cursor and select the target. The on-screen cursor appeared as a black dot and the targets appeared as red circles. We implemented a standard pointing task [1] where the next target with the appropriate distance and width appeared after the current selection. For selection, we used a keyboard and asked the users to hit the space-bar when they feel the black dot is in the red target. As is standard, we asked the users to perform the task as quickly and as accurately as possible. Participants were provided regular breaks.

5.5.3 Measures

We measure Movement Time (MT) – the time it took for the user to complete a trial, and Incorrect Trials (IT) – the number of incorrect trials out of five for each condition. An incorrect trial is when a user misses a target (i.e the space-bar is hit while the cursor is outside the target) or if the movement time is an outlier. We removed the incorrect trials from the movement time measurement.

5.5.4 Results

For each axis, we conducted 3-way repeated measures ANOVAs on movement time and incorrect trials. Greenhouse-Geisser corrections were applied for violations of sphericity. The results were similar for both axes. For both yaw and pitch, for Movement Time (MT) and Incorrect Trials (IT), we found significant main effects of all three Method, Distance, and Width. No interaction effects were found. Table 5.1 shows the main effects statistics.

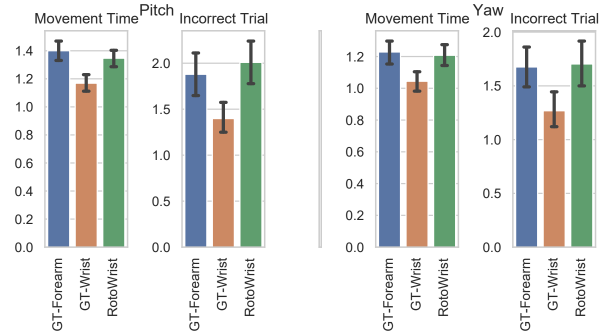
Pairwise comparisons with Bonferroni corrections (Table 5.1) showed that the effects of distance and width broadly adhered to Fitts' law with Movement Time increasing with increasing distance and decreasing with increasing width. Since there are no interaction effects, we focus on the effect of Method on Movement Time and Incorrect Trials.

Effect of Method

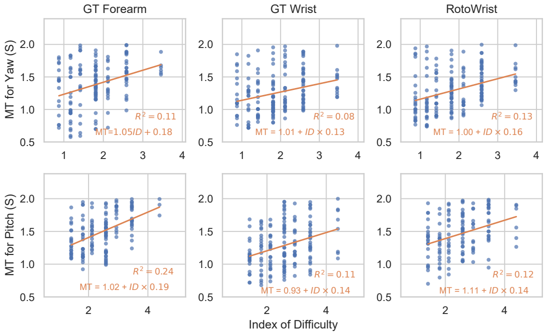
Figure 5.6a shows the movement time and incorrect trials for each method for both axes. Ground truth wrist is faster than ground truth forearm or RotoWrist according to the pairwise comparisons, although the difference is only 0.25s. This shows that users can complete the pointing task using RotoWrist across all distances and width while only being

Table 5.1: Pointing Study: 3-way ANOVA Main effects and pairwise comparisons. No interaction effects were found.

	Condition	**ANOVA**	**Pairwise Comparisons**
Yaw MT	Method	$F(2,22)$= $9.252, p < 0.005, \eta^2 = .457$	$GTW < GTF(p < .05), GTW < R(p < .05)$
	Distance	$F(2,22)$= $40.897, p$< $0.001, \eta^2$= $.788$	$50 < 100 < 200$ ($p < .05$) for all
	Width	$F(2,22)$= $48.817, p$< $0.001, \eta^2$= $.816$	$60 < 20(p < .001), 40 < 20(p < .001)$
Yaw IT	Method	$F(2,22)$= $7.558, p < 0.005, \eta^2 = .407$	$GTW < GTF(p < 0.005)$
	Distance	$F(2,22)$= $7.064, p < 0.005, \eta^2 = .391$	50< $200(p$< $.05), 100 < 200(p < .05)$
	Width	$F(1.29, 14.19)$=$8.54, p$<$0.01, \eta^2$=$.437$	$60 < 20(p < .05), 40 < 20(p < .05)$
Pitch MT	Method	$F(2,22)$= $13.034, p$< $0.001, \eta^2$= $.542$	$GTW < GTF(p < .005), GTW < R(p < .05)$
	Distance	$F(2,22)$= $34.115, p < 0.001, \eta^{=}.756$	$100 < 200 < 400(p < .05)$ for all
	Width	$F(2,22)$= $36.413, p$< $0.001, \eta^2$= $.768$	$60 < 40 < 20(p < .05)$ for all
Pitch IT	Method	$F(2,22)$= $6.862, p < 0.01, \eta^2 = .384$	$GTW < GTF(p < .05), GTW < R(p < .05)$
	Distance	$F(2,22)$= $11.004, p$<$0.001, \eta^2$=$.500$	100< $400(p$< $.05), 200$< $400(p$< $.05)$
	Width	$F(2,22)$= $19.947, p$< $0.001, \eta^2$= $.645$	$60 < 20(p < .005), 40 < 20(p < .005)$



(a) Mean and standard deviation of movement time and incorrect trials for pitch and yaw axis.



(b) Yaw and Pitch Movement Time vs Fitts' Index of Difficulty

Figure 5.6: Pointing evaluation results

slightly slower than ground truth wrist tracking and similar to ground truth forearm tracking. A similar trend is observed in incorrect trials.

Adherence to Fitts' Law

Figure 5.6b illustrates the movement time as a function of index of difficulty (ID) for the three different methods among all participants. A linear regression model was used to fit the data. All three methods show adherence to Fitts' law, albeit a weak one given the lower R^2 values. This shows that alternate modeling approaches should be investigated to better model pointing based on hand-tracking.

Overall, the results show that RotoWrist enables pointing tasks to be performed with performance that is only slightly below ground truth wrist tracking. Although the RotoWrist has similar performance in movement time and error rate to ground truth forearm tracking, it is still the preferred choice compared to ground truth forearm tracking since it requires less movement from the user and can be performed in a subtle, comfortable manner. This was profoundly echoed by the participants feedback where they mentioned more fatigue when performing ground truth forearm movements for pointing task. Using a wearable device like RotoWrist to track wrist pose is also easier from an implementation perspective than enabling 6-DoF forearm tracking, which requires another point of instrumentation in the world or on the head (e.g., HMD-mounted cameras to track the arm).

5.6 Applications

The ability to track one's wrist orientation reliably and precisely without per-session training enables a wide range of applications. We envision these applications to be with or without a head-mounted display. In this section we discuss and demonstrate a few features that RotoWrist can enable.

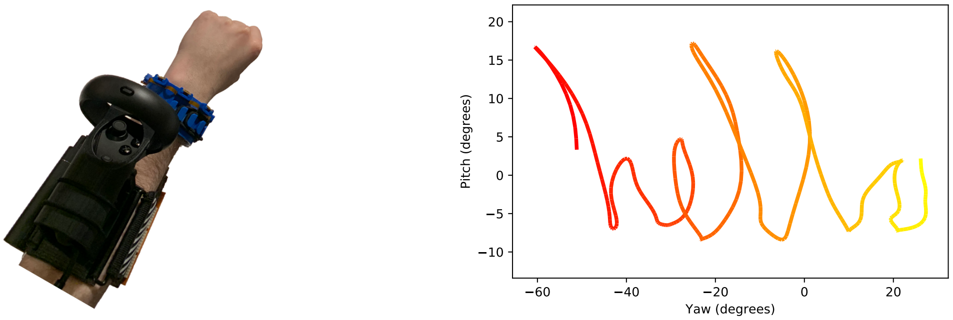
5.6.1 Pointing in AR/VR

Virtual and augmented reality platforms represent a promising direction for next-generation computing platforms. For many applications, being able to point at object precisely is of a great importance. The handheld controller is a useful input device that is common in VR, but moving the hands without the encumbrance of holding a physical handheld controller could lead to more natural and casual experiences.

To demonstrate the potential of RotoWrist as a pointing device, we demonstrate how it can be used to play Beat Saber[3], one of the most popular VR experiences. The game is normally controlled by two handheld controllers, and is particularly challenging as it requires precise targeting and low-latency.

---

[3] https://beatsaber.com/



(a) A Rift-S controller is added to the controller armband to facilitate 6-DoF tracking of the arm.

(b) A user drawing ”hello” in free space with their wrist movement while the wrist orientation is getting tracked with RotoWrist

Figure 5.7: Applications of RotoWrist

To build this demo, we attached a Rift-S[4] handheld controller to the top of the controller armband as shown in Figure 5.7a. The controller tracked the user’s arm pose with respect to the headset. During a simple calibration phase, we determine the transformation from the pose of the Rift-S controller to the pose of the hand. At runtime, we add the wrist angles estimated from RotoWrist to the transformed pose of the controller to accurately capture the 6-DoF pose of the hand.

We asked ten users to try our demo for one song and provide feedback. We purposefully had them play the game using a Rift-S handheld controller on the right hand while having our device on their left hand so that they can compare the two input devices. Almost all users were very excited about the idea of playing BeatSaber without a handheld controller and were impressed by the tracking accuracy. Two people mentioned that they felt that RotoWrist was more delayed than the Rift-S handheld controller but not enough to make them uncomfortable. Further improvements in speed and accuracy may be possible by

---

[4] https://www.oculus.com/rift-s/

using a Kalman filter to proactively estimate the next wrist orientation.

5.6.2 Free-form Hand Drawing

To highlight RotoWrist's ability to perform fine-grained continuous tracking (beyond gestural swipes), we implemented a drawing application driven solely by the user's wrist orientation. Since the wrist has smaller range of motion in radial/ulnar deviation, the drawing canvas shifts to the left by a few pixels each frame to enable an infinite canvas.This free-form drawing can be used to draw shapes for gestures recognition or can be used for handwriting reconstruction. Figure 5.7b shows a user that used RotoWrist to write ”hello” in the mid air. With appropriate visual feedback, we anticipate the performance to be even better.

5.6.3 Wrist Rotation Visualizer

RotoWrist can also be used to reconstruct the exact orientation of the wrist with respect to the arm.In virtual reality scenarios, a person can move only their wrist, and the virtual hand will also move the same way. Although we do not implement click events, it has been demonstrated that using index-thumb pinch can enable click events [64]. In this way, in mobile scenarios, RotoWrist system can be used instead of a handheld controller.

5.7 Conclusion

In this work, we demonstrated RotoWrist, a wearable wrist-worn device that performs continuous 2-degree-of-freedom (DoF) wrist tracking using ToF IR sensing modules. RotoWrist works across sessions and users without requiring supervised user calibrations. The RotoWrist system consists of a wristband that uses eight ToF IR modules to continuously measures the absolute distance from the hand in real-time. As the user moves their wrist, the relative distance between the hand and each sensor changes and collectively RotoWrist uses these measurements to track 2-DoF wrist orientation—flexion/extension and radial/ulnar deviation. RotoWrist offers a rich input source for a variety of wearable devices, including smartwatches and HMDs. With RotoWrist, a user can provide continuous input using their wrist, in eyes-away contexts where line-of-sight to a camera might be difficult to maintain.

Chapter 6

HAND TO HAND RELATIVE POSE TRACKING USING TWO RINGS

6.1 Introduction

Humans use both hands frequently to perform everyday actions. We naturally use our hands to perform tasks such as picking up an item, washing dishes or in more precise tasks such as hammering a nail to the wall. Typically, our manual operations can be divided into two types: unimaual (one-handed) [12, 150] and bimanual (two-handed) [81, 110]. The bimanual operations can be further categorized into symmetric, where both hands perform similar tasks and have the same level of importance; and asymmetric, in which the two hands have different roles at the same time. The ever growing requirements to offer more mobility has motivated researchers to design easy to use wearable devices and therefore ring worn devices like smart-watches and smart-rings have become increasingly mainstream.

Previous works have investigated the feasibility of using these devices to sense hand movements and gestures. The device is usually worn on one hand, either the dominant hand, for tracking hand movements [36, 26, 57, 139] or the non-dominant hand, for detecting touch events [157, 102]. However, few works have investigated the potential bimanual tracking that is created by combining the sensing capabilities from devices worn on the user's dominant and non-dominant hands. One of the most recent work in this space is by Lu et. al. where they explore the design space of hand-to-hand gestures by designing a group of gestures that are performed by touching one hand with the other hand using two off-the-shelf wrist-worn devices [70]. Although this is a great work and can be used to invoke commands in real-life contexts, it does not result in full tracking between the two hands. Although discrete gestures are useful, they are just one part of the interaction language needed for mobile wearable computing. For tasks such as drawing or object manipulation, an input device that supports continuous tracking will be most appropriate.

In this work, we present BiRing, an always available ring form factor input device for augmented and virtual reality devices. The BiRing system consists of a transmitter ring that continuously generates a small oscillating magnetic field and a receiver ring that measures the field and estimate the pose of transmitter ring with respect to it. We envision using the back of the non dominant hand as a canvas and the index finger of the dominant hand as the stylus. When the user's finger hover or touch the hand, we can compute the location of the finger, enabling real-time 2-DoF tracking on the back of the non dominant hand.

We envision BiRing to be used as a standalone input device or in tandem with a HMD. For standalone scenarios, users can provide text entry through hand drawing on the back of the non dominant hand. Our device can also be used in tandem with a HMD. On of the challenges in current VR/AR headset appears when a hand covers the other, or when the hands touch or hover above each other. In these natural scenarios, the tracking temporarily breaks using camera for hand tracking. Using BiRingand hand tracking enable tracking the absolute pose of the hand in head-space with respect to the user's vision in these scenarios. Collectively the two aforementioned systems enable many compelling applications. For example, buttons, sliders, and interactive widgets can be placed in the back of hand.

6.2 Implementation

Our system consists of two ring worn devices, one goes on the dominant hand and the other on the non-dominant hand as shown in Figure 6.1. The dominant ring-worn device generates magnetic fields and the receiver ring measures the resulting field. In the following of this section, we would first elaborate on the hardware and then talk about the tracking algorithm used.

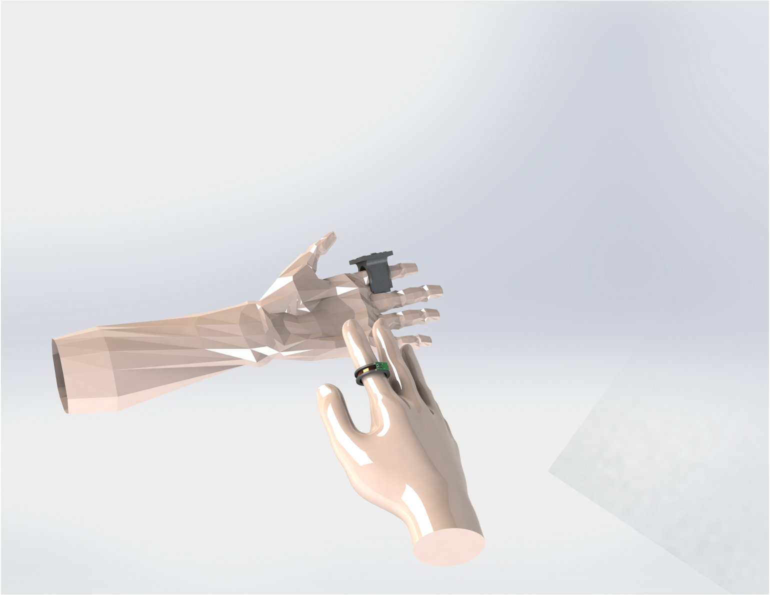


Figure 6.1: BiRingconsist of a transmitter ring that goes on the dominant hand and a receiver ring that goes on the non dominant hand.

6.2.1 Hardware

Ring-based Transmitter

In designing BiRing, we pursued a similar design to AuraRingwhere it resemble device form-factors that users are already accustomed to in order to present a plausible path forward for everyday use. For the ring-based transmitter, we used a single-axis low-profile transmitter coil that emits a magnetic field oscillating at $32.768\,\mathrm{kHz}$. The generator coils consists of 800 turns of 42 AWG magnet wire wound around a 3D-printed ABS ring with a diameter of $20\,\mathrm{mm}$. The inductance of the coils is approximately $15\,\mathrm{mH}$.

For the transmitter electronics, a small custom PCB ($6.9\,\mathrm{mm} \times 11.4\,\mathrm{mm}$), sits horizontally on top of the 3D printed ring. The transmitter uses a surface-mounted oscillator (ACZ-32.768) to generate a $32.768\,\mathrm{kHz}$ square wave. We use a capacitor network to tune the coil's impedance at $32.768\,\mathrm{kHz}$ in order to maximize the transmit power of AuraRing. The transmitter is powered with two $12\,\mathrm{mA\,h}$, $1.55\,\mathrm{V}$ coin cell batteries with a diameter of $4.8\,\mathrm{mm}$ and a height of $2.1\,\mathrm{mm}$.

Ring-based Receiver

The BiRingring-based receiver consist of a sensors and a controller boards.At a high level, the sensor board measure a magnetic field and the controller board convert the measurements to digital signals and communicate them to a host computer.

The sensor boards measure the magnetic field generated by the transmitter ring using an off-the-shelf three-axis orthogonal receiver coil (Grupo Premo 3DC15). The signal from each axis is fed to an amplifier (INA826) with a gain of $44\,\mathrm{dB}$. We then use a low-noise and low-voltage drop Schottky diode network (SMS7630) in a full-wave bridge rectifier configuration to demodulate each of the channels. We have used this passive configuration to demodulate the field in order to save power. It is worth mentioning that using this method, we only have access to the fields' magnitudes; BiRing does not know whether each channel is in- or out-of-phase with the transmitter. Because of the interaction space that we are interested in which is only on the back of the hand, most of the channels' signs remain constant, so there is not much information lost. In other words, if a channel is out-of the phase for a couple of seconds, it will remain out of the phase.

These magnitude signals are passed to a controller board with a 10-wire FFC cable where a SAR analog-to-digital converter (AD7265) samples the three differential signals simultaneously at $31.25\,\mathrm{kHz}$. Each channel on the controller board has an ADC driver (LTC6363) to achieve high precision, resolution, and throughput. Finally, the sampled data is passed to an ultra-low power MCU (MSP430FR2422) for further processing and communication.

6.2.2 Algorithm

We propose to use a computationally simple models to regress to a 2-DoF position of the dominant finger on the back side of the non dominant hand. We trained a computationally simple neural network model with a single hidden layer of 32 nodes to fit a function that maps the magnetic field vector to a 2-dimensional position vector. Training is performed using the Levenberg-Marquardt algorithm. At runtime, position estimation equates to two matrix multiplications (6×32 and 32×3), that can easily run on a mobile processor.

6.2.3 Data Collection Setup

We use a 8-camera[1] optical motion capture system to track the ground truth position and orientation of the transmitter and receiver rings. To facilitate this, we place IR retroreflective markers on both devices. The motion capture system reports the pose of the the two rings in room-space at $240\,\mathrm{Hz}$. Software in Python on a PC logs both the motion capture data and BiRing sensor data to disk for offline analysis. Finally, we synchronize the sensor stream from AuraRing ($472\,\mathrm{Hz}$) with the ground truth ring pose stream ($240\,\mathrm{Hz}$).

6.2.4 Calibration

In order to treat the measurements from the ADC as magnetic field strengths, a calibration procedure must be performed to account for things like sensitivity of each coil and the relative positioning of the sensor coil on the PCB. Altogether, they capture offsets between the motion capture positions and the effective magnetic origins as well as effects of the BiRing analog signal processing pipeline. We model these effects using the method that was validated for electromagnetic tracking in [90]. This calibration is intended to be performed only once per device, i.e. through a factory calibration step. Below, we briefly outline these parameters and how they fit into the BiRing sensor model.

Coordinate System Offsets

Due to manufacturing tolerances, the exact position of each sensor with respect to origin reported by motion capture may not exactly align with the computed values from the CAD

---

[1] OptiTrack Prime 17W

design. Consequently, we define offset positions and orientations to refine sensor pose in the receiver frame. These slack terms are expected to be small — on the order of a couple of millimeters or degrees.

Modeling Analog Signal Chain

We then use the dipole model introduced [90] to compute the magnetic field in the sensor reference frame. However, the sensors and amplitude demodulation pipeline do not represent a perfect measurement of the magnetic field. We also model the effects of noise and the diode forward voltage drop using a model that was validated for electromagnetic tracking in [142]. Altogether, this model includes 23 parameters that must be learned for the sensor coil.

Learning Sensor Model Parameters

To learn these parameters, we collected a dataset that captures a wide range of motion of the ring. One of the researchers wore the receiver ring while using the other hand to move and rotate the transmitter ring freely within in a volume around the hand. During this process, the two rings' pose are captured by the motion capture system. We took a random sample of 30k frames from this dataset to use for learning and formulated the problem as a non-linear optimization problem, which we solve using Ceres Solver [4]. In doing so, the solver seeks to find the parameter set which minimizes the difference between the observed sensor values and the estimated sensor values, computed according to the observed pose and the sensor model. The solver uses the Levenberg-Marquardt trust region algorithm for minimization. Upon convergence, this results in a set of parameters that can be used to estimate the sensor measurements as a function of pose.

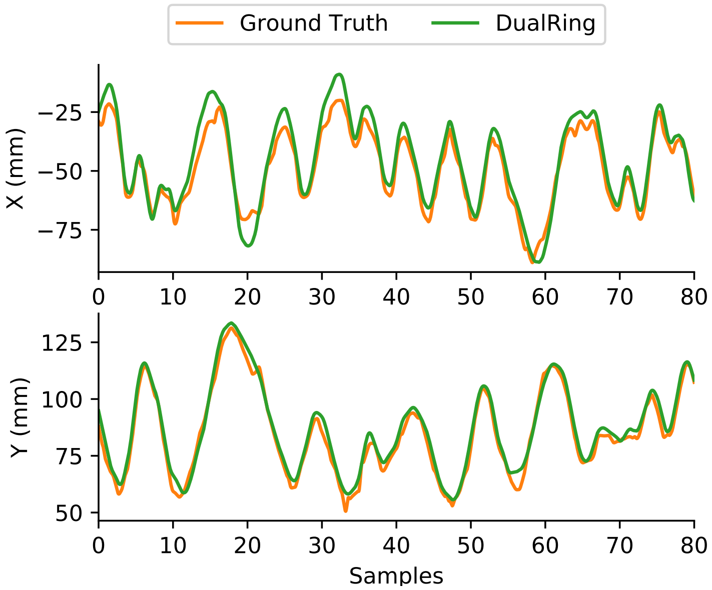


Figure 6.2: A representative few seconds of 2D orientation tracking for one of the participants (mean error for cross-user: 4 mm

6.3 *User Evaluation*

In this section, we evaluate BiRing's performance in 2-DoF relative tracking between two rings.

6.3.1 *Participants and Apparatus*

We recruited 10 participants (8 males, 2 females) from the local campus. The data collection was carried out in the same 8-camera motion capture lab used to calibrate the sensor model. Upon arrival, we asked the participants to sit on a chair while having the back of their left hand facing themselves. A researcher helped the participants put on the transmitter ring on their right index finger and the receiver ring on the left index finger as shown in Figure 6.1.

The evaluation was conducted in two sessions. In each session participants were asked to perform 2 minutes of free-form movements of the index finger on and above the back

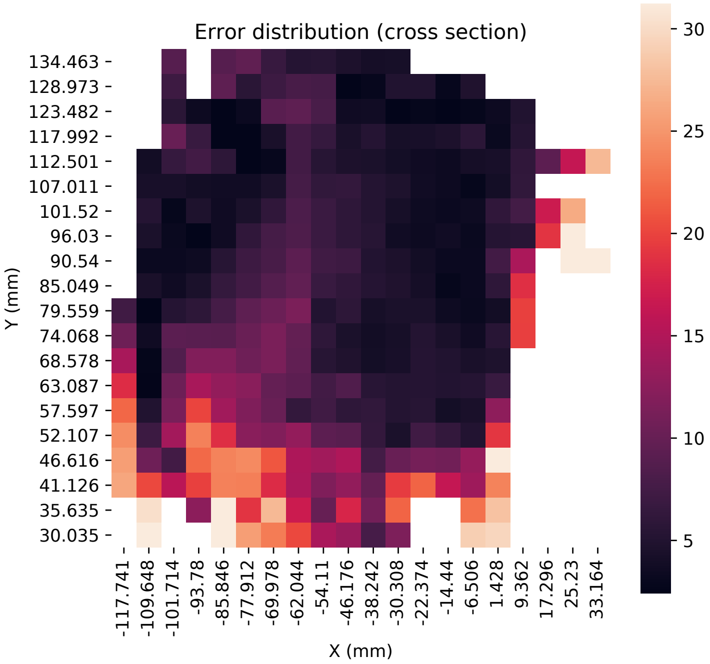


Figure 6.3: Heatmap showing spatial distribution of error among all users for cross-user evaluation.

of the other hand while being asked to move to all possible locations on the back of the hand. After the first session, the transmitter ring was taken completely off. The users were instructed to take a rest. After the break, the researcher asked the participants to put the transmitter ring back on, and the same procedure was given to the user. During this time, the ground-truth pose and the magnetic sensors data was recorded by a Python program.

6.3.2 Result

As stated in Section 1, a widespread input solution should be robust and extensible (e.g. supporting different users and contexts out of the box). Therefore, in this section, we evaluate how our tracking holds between users. Notably, we train a model from the first session of all participants except one and test on all of the data from the left out user. We repeat

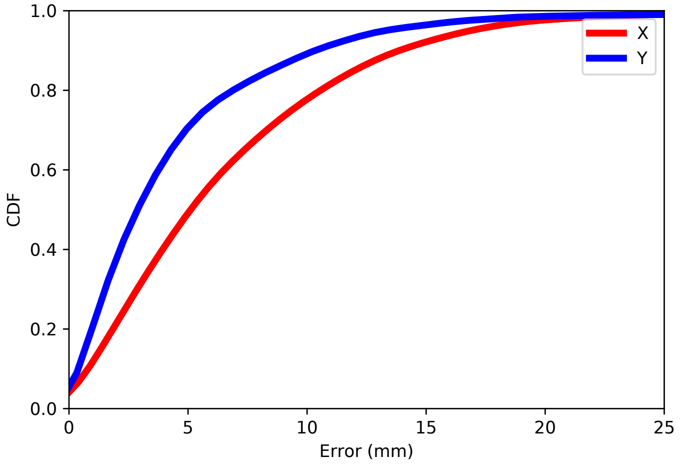


Figure 6.4: Cumulative distribution function of x and y tracking error.

this for all of the participants. The measured RMSE for x and y across users is $6.5\,\mathrm{mm}$ and $4.5\,\mathrm{mm}$ respectively.

A representative few seconds of 2DoF tracking for one of the participants is shown in Figure 6.2. Figure 6.4 shows the cumulative distribution function (CDF) of tracking accuracy cross users. Figure 6.3 shows the spatial distribution of orientation tracking error of the cross-user model projected onto a 2D plane among all users.

Chapter 7

CONCLUSION AND DISCUSSION

In this thesis, I showed how by using data-driven approaches, we can build a more precise, subtle and computationally simpler inputs to the wearable devices.

In chapter three, we demonstrate a 6-DoF tracking system capable of tracking position with a median error of $5.5\,\mathrm{mm}$ and a median orientation error of $0.8°$ within arm's reach around the head while using less than $50\,\mathrm{mW}$ on the controller. This approaches the performance of commercial electromagnetic tracking systems, such as the Polhemus G4, while using an order of magnitude less power and allowing optimization for form-factor. One of the advantages of Aura system over other electromagnetic tracking systems is the ability to use arbitrary transmitter coil configurations. While we designed the Aura prototype as a snap-on device, the transmitter coils could also be placed directly onto a PCB behind the display of a VR system or embedded within the frames of a pair of glasses. By reducing the dependence on orthogonal dipole models, we widen the design space for head-mounted computing systems with tracked devices.

In chapter four, we demonstrated AuraRing, a wearable device that enables precise tracking of the pose of a smart ring with respect to the wristband. The use of short-range electromagnetic tracking offers the unparalleled precision of EM tracking while minimizing the effects of environmental interference. In this chapter, we have proposed two methods for tracking. For more precise tracking we use an iterative model that achieve a position accuracy of $4.4\,\mathrm{mm}$ and a 2-DoF mean orientation accuracy of $4.65°$. In mobile scenarios where computation power is at a premium, we propose a relatively small neural network model with only one hidden layer. Using this approach AuraRingis able to achieve an accuracy of $6\,\mathrm{mm}$ across all users. We have also shown that for achieving even better accuracy, a per-session recalibration can reduce the position and orientation error to $1.5\,\mathrm{mm}$ and $1.7°$ respectively.

Among related work in sensor-based tracking of the fingers, AuraRing is most similar to Finexus [16]. In AuraRing, we prioritized form-factor and designed the system from the ground up with wearability in mind—both in the design of the ring-worn transmitter and the wrist-based sensors. AuraRing uses significantly less power ($2.3\,\mathrm{mW}$ on the finger), uses an untethered ring, and estimates 2-DoF rotation in addition to position. In terms of accuracy, AuraRing's $1.5\,\mathrm{mm}$ single-session accuracy is comparable to that reported in the single-session evaluation of Finexus. Moreover, the AuraRing accuracy holds at typical finger-wrist distances, which exceed the operating range of Finexus.

In chapter five, we demonstrated RotoWrist, a wearable wrist-worn device that performs continuous 2-degree-of-freedom (DoF) wrist tracking using ToF IR sensing modules. RotoWrist works across sessions and users without requiring supervised user calibrations. The RotoWrist system consists of a wristband that uses eight ToF IR modules to continuously measures the absolute distance from the hand in real-time. As the user moves their wrist, the relative distance between the hand and each sensor changes and collectively RotoWrist uses these measurements to track 2-DoF wrist orientation—flexion/extension and radial/ulnar deviation. RotoWrist offers a rich input source for a variety of wearable devices, including smartwatches and HMDs. With RotoWrist, a user can provide continuous input using their wrist, in eyes-away contexts where line-of-sight to a camera might be difficult to maintain.

In chapter six, we showed how BiRing can be used to perform continuous 2DoF tracking of the dominant hand on the back side of the non dominant hand. BiRing works across users without any user calibration.

7.1 Complementing Vision-based Tracking

We envision AuraRing, RotoWrist, and BiRing be use either with or without a head-mounted display. For mixed reality scenarios, head-mounted cameras are becoming increasingly capable of tracking the hands and fingers, but doing so comes at a significant power and compute cost. Tracking IR fiducial markers in a wristband is a much simpler task. However, in cases where full five-finger tracking is desired, AuraRing and BiRing can serve as a robust prior to constrain the search space for a traditional hand-tracking algorithm. For interaction with other wearable devices, AuraRing and BiRingcan be used as an indirect input device.

Furthermore, even though head-mounted cameras are getting better at tracking hands and fingers, they still require line-of-sight. In contrast, tracking a wristband with infrared fiducial markers is a much simpler task. The combination of RotoWrist with a wristband tracker that results in a complete 6-DoF tracking system, may be preferable over a hand-held controller in some scenarios. For example, a user could comfortably interact with their hands at their side where computer vision might fail, then seamlessly move into the field-of-view of the cameras, which would begin tracking the arm or wristband pose, and start direct manipulations. As a standalone device, RotoWrist offers a rich input source for smartwatches. With RotoWrist, a user can provide input using their wrist while keeping their arm motionless at their side or resting on a table. With some modifications, we envision that RotoWrist can capture information about the pose of the fingers as well, e.g., detecting a fist vs a full extended hand for target selection. This could be useful in mid-air drag & drop interactions.

7.2 Extensions, Limitations, and Future Work

While the approach taken in Chapter 3 demonstrates the feasibility of precise tracking, it does not fully account for all possible sensor positions and orientations. Additional robustness can be achieved by collecting data in all possible configurations, perhaps with the use of a robotic arm. For manufacturing purposes, a separate, externally calibrated sensor can be used to train the magnetic field models, as these depend only on the magnetic field, not on any specific measurement of the Aura device.

Additional performance can likely be achieved by optimizing the placement and shape of the transmitter coils in Aura and the placement of the reciever coils in AuraRing. The Aura coils were designed for the form-factor of a particular HMD, but our simulation results reveal a performance difference between a dipole model and a model of our coils. Leveraging simulation and optimization tools, we expect one can optimize the design for a particular use case. Despite this, the Aura prototype demonstrates reasonable tracking performance without any iteration over transmit coil design. AuraRing performance could also further be improved by additional optimization of the sensor coil positions on the wristband. In general, the performance will improve as they are more spread out. Since we envisioned the ring to be worn on the index finger which is more close to the inside of the hand, we placed two sensors to the left side of the wristband, assuming right-handed use. AuraRing's hardware has the ability to support four sensor boards without modification of the electronics. For additional performance or for left-handed use, one could install another sensor board to the right side of the wristband.

With some modifications, AuraRing can support tracking of multiple rings at the same time for full hand tracking. Currently the wristband is tuned to the transmitter frequency, but by tuning each ring to different frequencies, AuraRing can time multiplex between the ring's frequencies and measure the resulting fields. AuraRing has a maximum sensing range of $30\,\mathrm{cm}$. For multiple users in the same environment, there is likely little that needs to be done. However, if users wish to operate multiple devices in close proximity, different frequencies can be used.

While both Aura and AuraRing was designed with low-power operation in mind, we expect that additional engineering improvements can further reduce power consumption. For example, additional coil windings on the transmitter and the use of a fixed oscillator instead of a programmable function generator in Aura can save significant power on the head-mounted transmitter.

In this thesis tracking models were trained using data collected from an optical motion capture system. However, for researchers who wish to use our system for their own projects, we anticipate that training can also be performed using commodity low-cost trackers, such as the HTC VIVE Tracker.

A challenge in the design of any smart ring is sizing it for an appropriate fit across users. For the purposes of the study done in chapter four, we made the ring larger than average and used tape to wrap around the finger to ensure a snug fit. A more robust design might use several fixed size coils (e.g. small, medium, large) with replaceable insets in standard ring sizes.

Although Chapter4 focuses on the use of a ring to track the finger, AuraRing could also be used as a generic tracking platform for handheld objects. For example, by putting a coil around a stencil or pen, the object is now trackable with respect to the wrist. Pen input could be automatically digitized, a toy could be automatically turned into a gaming controller, or a toothbrush can monitor how someone is brushing their teeth.

RotoWrist can be generalized across sessions and users. We will now discuss how RotoWrist can be generalized across different palm posture, location on the forearm and rigidity. Although in our evaluations, we asked the participants to make a fist, we observed that the tracking holds even as the user opens their fist and starts moving their fingers. It's worth mentioning that the tracking error increases as the user moves their thumb or pinky finger outwards, making a motion that, to RotoWrist, appears similar to a radial/ulnar deviation motion. Future work should consider how additional modeling or the use of additional sensors can distinguish between finger motion and wrist motion.

Although RotoWrist achieves a cross-user RMSE of 5.9° and 6.8° for pitch and yaw, respectively, our system can be vulnerable to major changes in the location of the band. We expect this can be addressed by using a few seconds of online learning. Furthermore, a challenge in the design of any wristband is sizing it for an appropriate fit across users. We also wanted the wristband to be rigid in order for the ToF modules to be roughly at the same pose for all users. Incorporating these modules in a flexible band requires more attention as shifts in the position of the modules would degrade performance. Furthermore, the relative angle of each module with respect to the band is an important factor to consider. While incorporating RotoWrist in a flexible band, care should be take so that the modules do not change orientation.

RotoWrist consists of the sensing wristband with eight ToF IR modules and a controller arm band that supplies powering the sensing wristband and handles data communicating back to a host PC. Here, we have focused on the design of the sensing wristband and optimizing the placement of the ToF IR modules. Incorporating the controller components into the wristband is a fairly straightforward task—most smart watches do have an MCU and Bluetooth. In designing the sensing wristband, we placed the ToF modules evenly around the wristband. RotoWrist's performance could be improved further by additional optimization of the number and placement of ToF modules on the wristband. Future work could also investigate tracking accuracy with fewer sensors.

If RotoWrist is combined with other tracking systems, e.g., an HMD-based hand tracker, the additional calibration step discussed in Section 5.4.3 could be done automatically during the first few seconds of use. After this, the hand tracking from the headset can stop and RotoWrist will be used for tracking.

7.3 Future Directions

7.3.1 Sensor Fusion

In this thesis I have focused on each device or sensing technique separately, i.e. tracking the orientation of the wrist using IR, or tracking the finger using electromagnetic tracking. However, significant advantages can be realized by combining different sensing technique from different sources of information. For example, in RotoWrist we proposed an optically head-based wrist tracker and the use of IR for wrist angle tracking. With both sources of information, tracking the wrist with respect to the head is a simple coordinate system transformation. As of another example, Aura can have better performance if it included integration with an onboard IMU. In a production-grade system, one would use Kalman filtering techniques to fuse the electromagnetic pose estimate with inertial measurements to improve speed and precision.

7.3.2 Power consumption

In this thesis, I have explored sensing modalities and have designed hardware solutions that particularly result in a tracking system that have low power consumption and can be operated by a small battery. A great follow up to this thesis can be done by exploring how one can make these sensors passive or battery-free. For example, future work can also explore the use of a completely battery-free ring. The AuraRing system uses a transmitter on the ring and measure the resulting fields on the wristband. An alternative method of tracking could be to use AuraRing's wristband's coils as both the transmitter and receiver. In this topology, the ring is a tuned LC network that will change the measurements on the wristband based on its pose.

7.3.3 Calibration

In this thesis, I have prioritized robustness and accuracy and designed the system from ground up to support continuous tracking with minimal to no calibration.

In Chapter 4, the use of a physics and sensor model is a significant advantage of tracking algorithm, compared to other methods. It allows simulation and experimentation to optimize system configuration and decouples model training from user evaluation. That said, AuraRing does still require the use of a one-time factory calibration. For the prototype presented in Chapter 4, such a calibration is essential, due to the manufacturing tolerances of the device assembly. However, it is possible that for a mass-produced device, the tolerances can be made tight enough such that a single model would suffice for all units.

Although the intend for the factory calibration is to be used across all sessions, presented results showed that tweaking the calibration model using some per-session data can reduce the tracking error to 1.5 mm. We attribute the change in calibration parameters between sessions to electrical effects in the signal processing chain. Several components in the pipeline, particularly the rectifier diodes, are sensitive to environmental temperature fluctuations. We expect that with more attention paid to analog stability, the global error will approach the per-session error. Another promising approach is migrating to digital processing for demodulation. This could enable reconstruction of both the magnitude and phase of the signals. We leave such explorations to future work.

By modeling the magnetic fields empirically, Aura can operate near ferromagnetic materials, such as those found within a head-mounted display. However, this approach accounts only for static distortions to the field. Large metallic objects brought nearby the device will degrade tracking performance and the calibration may be invalidated at runtime. Still, because Aura, AuraRing, and BiRing is inside-out tracking systems, metallic objects must be near the head or hand for distortions to occur. In such situations, one cannot simply ask the user to recalibrate. Instead, future research should explore sensor fusion techniques for online calibration. Future work can explore techniques to fuse electromagnetic tracking with optical or inertial tracking to maintain accuracy in the presence of nearby distortions. Dynamic distortions caused by particular electronics within the display or wristband will likely be localized to a particular frequency and can be eliminated by carefully choosing the frequency of operation. Another method is to use a head-mounted camera which track a fiducial on the controller that is only visiblesometimes and only provides partial information about pose

7.4 Additional document

To allow other researchers better understand and leverage my work, I have open-sourced the hardware designs and algorithms used in this thesis for Chapters 3, 4, and 6. The files are located in the footnote for Aura[1], AuraRing[2], BiRing[3].

---

[1] https://github.com/ubicomplab/Aura

[2] https://github.com/ubicomplab/AuraRing

[3] https://github.com/ubicomplab/BiRing

BIBLIOGRAPHY

[1] ISO, 9421-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. International Organization for Standardization. 2000.

[2] 5DT. 5DT Data Glove Ultra Series. `http://www.5dt.com/downloads/dataglove/ultra/5DTDataGloveUltraDatasheet.pdf`. Accessed: 2019-05-09.

[3] Javid Abbasov, Tom Horak, and Raimund Dachselt. Smartwatch-based pointing interaction. In Raimund Dachselt and Gerhard Weber, editors, *Mensch und Computer 2018 - Tagungsband*, Bonn, 2018. Gesellschaft für Informatik e.V.

[4] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[5] Christoph Amma, Marcus Georgi, and Tanja Schultz. Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors. In *2012 16th International Symposium on Wearable Computers*, pages 52–59. IEEE, 2012.

[6] Riku Arakawa, Azumi Maekawa, Zendai Kashino, and Masahiko Inami. *Hand with Sensing Sphere: Body-Centered Spatial Interactions with a Hand-Worn Spherical Camera*. Association for Computing Machinery, New York, NY, USA, 2020.

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

[8] Teo Babic, Harald Reiterer, and Michael Haller. Pocket6: A 6DoF controller based on a simple smartphone application. In *SUI'18: 6th ACM Symposium on Spatial User Interaction*, pages 2–10, 2018.

[9] Yohan Baillot, L Davis, and J Rolland. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augumented reality*, page 67, 2001.

[10] Idil Bostan, Oğuz Turan Buruk, Mert Canat, Mustafa Ozan Tezcan, Celalettin Yurdakul, Tilbe Göksun, and Oğuzhan Özcan. Hands as a controller: User preferences

for hand specific on-skin gestures. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, page 1123–1134, New York, NY, USA, 2017. Association for Computing Machinery.

[11] W. Buxton and B. Myers. A study in two-handed input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, page 321–326, New York, NY, USA, 1986. Association for Computing Machinery.

[12] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3403–3414, New York, NY, USA, 2016. Association for Computing Machinery.

[13] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology*, UIST '15, page 549–556, New York, NY, USA, 2015. Association for Computing Machinery.

[14] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y Chen, Wen-Huang Cheng, and Bing-Yu Chen. Fingerpad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 255–260. ACM, 2013.

[15] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. Utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 237–244, New York, NY, USA, 2013. Association for Computing Machinery.

[16] Ke-Yu Chen, Shwetak N. Patel, and Sean 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, page 1504–1514, New York, NY, USA, 2016. Association for Computing Machinery.

[17] Wei-Tung Chen and Ling-Jyh Chen. Pokeball: A 3D positioning system using magnetism. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on*, pages 719–726. IEEE, 2017.

[18] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. Claw: A multi-functional handheld haptic controller for grasping, touching, and triggering in virtual

reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 654:1–654:13, New York, NY, USA, 2018. ACM.

[19] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. CLAW: A multifunctional handheld haptic controller for grasping, touching, and triggering in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 654. ACM, 2018.

[20] Catherine C. Cooksey and David W. Allen. Reflectance measurements of human skin from the ultraviolet to the shortwave infrared (250 nm to 2500 nm). In G. Charmaine Gilbreath and Chadwick Todd Hawley, editors, *Active and Passive Signatures IV*, volume 8734, pages 152 – 160. International Society for Optics and Photonics, SPIE, 2013.

[21] F. Cordella, F. Di Corato, G. Loianno, B. Siciliano, and L. Zollo. Robust pose estimation algorithm for wrist motion tracking. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3746–3751, Nov 2013.

[22] CTRL-Labs. CTRL-Labs. `https://www.ctrl-labs.com/`. Accessed: 2019-04-05.

[23] Houde Dai, Shuang Song, Chao Hu, Bo Sun, and Zhirong Lin. A novel 6D tracking method by fusion of 5D magnetic tracking and 3D inertial sensing. *IEEE Sensors Journal*, PP:1–1, 10 2018.

[24] Houde Dai, Shuang Song, Xianping Zeng, Shijian su, Mingqiang Lin, and Max Q.-H. Meng. 6D electromagnetic tracking approach using uniaxial transmitting coil and tri-axial magneto-resistive sensor. *IEEE Sensors Journal*, PP:1–1, 12 2017.

[25] Michael Deering. High resolution virtual reality. *SIGGRAPH Comput. Graph.*, 26(2):195–202, July 1992.

[26] Artem Dementyev and Joseph A. Paradiso. Wristflex: Low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 161–166, New York, NY, USA, 2014. ACM.

[27] T. Deyle, S. Palinko, E. S. Poole, and T. Starner. Hambone: A bio-acoustic gesture interface. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 3–10, Oct 2007.

[28] David Dobbelstein, Philipp Henzler, and Enrico Rukzio. Unconstrained pedestrian navigation based on vibro-tactile feedback around the wristband of a smartwatch.

In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 2439–2445, New York, NY, USA, 2016. ACM.

[29] Facebook. Oculus Quest Fact Sheet. `https://www.oculus.com/quest/features/?locale=en_US`. Accessed: 2018-09-39.

[30] Eric Foxlin, Michael Harrington, and George Pfeifer. Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *SIGGRAPH*, 1998.

[31] Marvin P Fried, Jonathan Kleefield, Harsha Gopal, Edward Reardon, Bryan T Ho, and Frederick A Kuhn. Image-guided endoscopic surgery: results of accuracy and performance in a multicenter clinical study using an electromagnetic tracking system. *The Laryngoscope*, 107(5):594–601, 1997.

[32] X. Ge, D. Lai, X. Wu, and Z. Fang. A novel non-model-based 6-DOF electromagnetic tracking method using non-iterative algorithm. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5144–5117, Sept 2009.

[33] Peregrine Glove. Peregrine Glove ST. `https://peregrineglove.com/products/peregrine-glove-st-full-kit-w-pod`. Accessed: 2019-05-09.

[34] Jun Gong, Da-Yuan Huang, Teddy Seyed, Te Lin, Tao Hou, Xin Liu, Molin Yang, Boyu Yang, Yuhan Zhang, and Xing-Dong Yang. Jetto: Using lateral force feedback for smartwatch interactions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 426:1–426:14, New York, NY, USA, 2018. ACM.

[35] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang "Anthony" Chen, Xiaojun Bi, and Xing-Dong Yang. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, 2018. Association for Computing Machinery.

[36] Jun Gong, Xing-Dong Yang, and Pourang Irani. Wristwhirl: One-handed continuous smartwatch input using wrist gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 861–872, New York, NY, USA, 2016. ACM.

[37] Jun Gong, Yang Zhang, Xia Zhou, and Xing-Dong Yang. Pyro: Thumb-tip gesture recognition using pyroelectric infrared sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 553–563. ACM, 2017.

[38] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. Qwertyring: Text entry on physical surfaces using a ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), dec 2020.

[39] Manoj Gulati, Farshid Salemi Parizi, Eric Whitmire, Sidhant Gupta, Shobha Sundar Ram, Amarjeet Singh, and Shwetak N. Patel. Capharvester: A stick-on capacitive energy harvester using stray electric field from ac power lines. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3):110:1–110:20, September 2018.

[40] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[41] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, page 3–12, New York, NY, USA, 2010. Association for Computing Machinery.

[42] Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 283–292, New York, NY, USA, 2011. Association for Computing Machinery.

[43] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. Understanding palm-based imaginary interfaces: The role of visual and tactile cues when browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 889–898, New York, NY, USA, 2013. Association for Computing Machinery.

[44] Chris Harrison and Scott E Hudson. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 121–124. ACM, 2009.

[45] Chris Harrison, Desney Tan, and Dan Morris. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 453–462, New York, NY, USA, 2010. Association for Computing Machinery.

[46] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[47] Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4203–4215, New York, NY, USA, 2016. ACM.

[48] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. Fingertrak: Continuous 3d hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(2), June 2020.

[49] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. Digitspace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1526–1537. ACM, 2016.

[50] Yasha Iravantchi, Mayank Goel, and Chris Harrison. Beamband: Hand gesture sensing with ultrasonic beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 15:1–15:10, New York, NY, USA, 2019. ACM.

[51] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. Interferi: Gesture sensing using on-body acoustic interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 276:1–276:13, New York, NY, USA, 2019. ACM.

[52] Mohd Noor Islam and Andrew J Fleming. Resonance-enhanced coupling for range extension of electromagnetic tracking systems. *IEEE Transactions on Magnetics*, 54(4):1–9, 2018.

[53] P. Jung, G. Lim, S. Kim, and K. Kong. A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors. *IEEE Transactions on Industrial Informatics*, 11(2):485–494, April 2015.

[54] Frederic Kerber, Pascal Lessel, and Antonio Krüger. Same-side hand interactions with arm-placed devices using emg. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, page 1367–1372, New York, NY, USA, 2015. Association for Computing Machinery.

[55] Wolf Kienzle and Ken Hinckley. Lightring: Always-available 2d input on any surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 157–160, New York, NY, USA, 2014. Association for Computing Machinery.

[56] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. Electroring: Subtle pinch and touch detection with a ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[57] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 167–176, New York, NY, USA, 2012. Association for Computing Machinery.

[58] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM, 2012.

[59] Wooyoung Kim, Jihoon Song, and Frank C Park. Closed-form position and orientation estimation for a three-axis electromagnetic tracking system. *IEEE Transactions on Industrial Electronics*, 65(5):4331–4337, 2018.

[60] Volodymyr V Kindratenko. A survey of electromagnetic position tracker calibration techniques. *Virtual Reality*, 5(3):169–182, 2000.

[61] Volodymyr V Kindratenko and William R Sherman. Neural network-based calibration of electromagnetic tracking systems. *Virtual Reality*, 9(1):70–78, 2005.

[62] Jack B Kuipers. SPASYN-an electromagnetic relative position and orientation tracking system. *IEEE Transactions on Instrumentation and Measurement*, 29(4):462–466, 1980.

[63] Katja M Langen, Twyla R Willoughby, Sanford L Meeks, Anand Santhanam, Alexis Cunningham, Lisa Levine, and Patrick A Kupelian. Observations on real-time prostate gland motion using electromagnetic tracking. *International Journal of Radiation Oncology* Biology* Physics*, 71(4):1084–1090, 2008.

[64] Gierad Laput, Robert Xiao, and Chris Harrison. Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 321–333, New York, NY, USA, 2016. Association for Computing Machinery.

[65] Claus Falck Larsen, Finn K. Mathiesen, and Steen Lindequist. Measurements of carpal bone angles on lateral wrist radiographs. *The Journal of Hand Surgery*, 16(5):888 – 893, 1991.

[66] Magic Leap. Magic Leap Fact Sheet. `https://www.magicleap.com/static/magic-leap-fact-sheet.pdf`. Accessed: 2019-04-05.

[67] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. Dualring: Enabling subtle and expressive hand interaction with dual imu rings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(3), sep 2021.

[68] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Trans. Graph.*, 35(4), July 2016.

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

[70] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(1), mar 2020.

[71] Kent Lyons. 2d input for virtual reality enclosures with magnetic field sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 176–183. ACM, 2016.

[72] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human–Computer Interaction*, 7(1):91–139, 1992.

[73] Jess McIntosh, Asier Marzo, and Mike Fraser. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. pages 593–597, 10 2017.

[74] Scott D. McPhee. Functional Hand Evaluations: A Review. *American Journal of Occupational Therapy*, 41(3):158–163, 03 1987.

[75] Michael Meehan, Brent Insko, Mary Whitton, and Frederick P. Brooks, Jr. Physiological measures of presence in stressful virtual environments. *ACM Trans. Graph.*, 21(3):645–652, July 2002.

[76] Michael Meehan, Brent Insko, Mary Whitton, and Frederick P Brooks Jr. Physiological measures of presence in stressful virtual environments. *ACM Transactions on Graphics (TOG)*, 21(3):645–652, 2002.

[77] CGM Meskers, HM Vermeulen, JH De Groot, FCT Van der Helm, and PM Rozing. 3D shoulder position measurements using a six-degree-of-freedom electromagnetic tracking device. *Clinical biomechanics*, 13(4):280–292, 1998.

[78] Microsoft. HoloLens 2 Fact Sheet. `https://www.microsoft.com/en-us/hololens/hardware`. Accessed: 2019-02.

[79] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[80] Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. Eyering: a finger-worn input device for seamless interactions with our surroundings. In *Proceedings of the 4th Augmented Human International Conference*, pages 13–20. ACM, 2013.

[81] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 177–186, New York, NY, USA, 2011. Association for Computing Machinery.

[82] Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota. 3D localization for sub-centimeter sized devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 108–119. ACM, 2018.

[83] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath 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, page 1515–1525, New York, NY, USA, 2016. Association for Computing Machinery.

[84] Daniel Natapov, Steven J. Castellucci, and I. Scott MacKenzie. Iso 9241-9 evaluation of video game controllers. In *Proceedings of Graphics Interface 2009*, GI '09, page 223–230, CAN, 2009. Canadian Information Processing Society.

[85] NDI. Medical Aurora - Medical. `https://www.ndigital.com/medical/products/aurora/`. Accessed: 2019-05-09.

[86] Uran Oh, Lee Stearns, Alisha Pradhan, Jon E. Froehlich, and Leah Findlater. Investigating microinteractions for people with visual impairments and the potential role of on-body interaction. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, page 22–31, New York, NY, USA, 2017. Association for Computing Machinery.

[87] Rohit Pandey, Pavel Pidlypenskyi, Shuoran Yang, and Christine Kaeser-Chen. Egocentric 6-DoF tracking of small handheld objects. *CoRR*, abs/1804.05870, 2018.

[88] S. Panëels, M. Anastassova, S. Strachan, S. P. Van, S. Sivacoumarane, and C. Bolzmacher. What's around me? multi-actuator haptic feedback on the wrist. In *2013 World Haptics Conference (WHC)*, pages 407–412, April 2013.

[89] E. Paperno, I. Sasada, and E. Leonovich. A new method for magnetic position and orientation tracking. *IEEE Transactions on Magnetics*, 37:1938–1940, July 2001.

[90] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. Auraring: Precise electromagnetic finger tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), December 2019.

[91] Valter Pasku, Alessio De Angelis, Guido De Angelis, Darmindra D Arumugam, Marco Dionigi, Paolo Carbone, Antonio Moschitta, and David S Ricketts. Magnetic field-based positioning systems. *IEEE Communications Surveys & Tutorials*, 19(3):2003–2017, 2017.

[92] D Perie, AJ Tate, PL Cheng, and GA Dumas. Evaluation and calibration of an electromagnetic tracking device for biomechanical analysis of lifting tasks. *Journal of biomechanics*, 35(2):293–297, 2002.

[93] G. Pirkl, K. Stockinger, K. Kunze, and P. Lukowicz. Adapting magnetic resonant coupling based relative positioning technology for wearable activitiy recogniton. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 47–54, Sept 2008.

[94] Thammathip Piumsomboon, Adrian J. Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction - INTERACT 2013 - 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part II*, volume 8118 of *Lecture Notes in Computer Science*, pages 282–299. Springer, 2013.

[95] Anton Plotkin and Eugene Paperno. 3-D magnetic tracking of a single subminiature coil with a large 2-d array of uniaxial transmitters. *IEEE Transactions on Magnetics*, 39(5):3295–3297, 2003.

[96] Polhemus. Polhemus G4. `https://polhemus.com/motion-tracking/all-trackers/g4`. Accessed: 2018-12-11.

[97] L. Quéval. BSmag toolbox user manual. Technical report, Dept. Elect. Eng., University of Applied Sciences Düsseldorf, Düsseldorf, Germany, April 2015. Accessed: 2018-12-11.

[98] Frederick H Raab, Ernest B Blood, Terry O Steiner, and Herbert R Jones. Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic systems*, (5):709–718, 1979.

[99] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.*, 43(1):1–54, January 2015.

[100] J. Rekimoto. Gesturewrist and gesturepad: unobtrusive wearable interaction devices. In *Proceedings Fifth International Symposium on Wearable Computers*, pages 21–27, Oct 2001.

[101] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, page 1093–1096, New York, NY, USA, 2011. Association for Computing Machinery.

[102] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. Synchrowatch: One-handed synchronous smartwatch gestures using correlation and magnetic sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), January 2018.

[103] D. Roetenberg, P. J. Slycke, and P. H. Veltink. Ambulatory position and orientation tracking fusing magnetic and inertial sensing. *IEEE Transactions on Biomedical Engineering*, 54(5):883–890, May 2007.

[104] Jaiyoung Ryu, William P. Cooney, Linda J. Askew, Kai-Nan An, and Edmund Y.S. Chao. Functional ranges of motion of the wrist joint. *The Journal of Hand Surgery*, 16(3):409 – 419, 1991.

[105] Farshid Salemi Parizi, Wolf Kienzle, Eric Whitmire, Aakar Gupta, and Hrvoje Benko. Rotowrist: Continuous infrared wrist angle tracking using a wristband. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21, New York, NY, USA, 2021. Association for Computing Machinery.

[106] Eva-Lotta Sallnäs, Kirsten Rassmus-Gröhn, and Calle Sjöström. Supporting presence in collaborative environments by haptic force feedback. *ACM Trans. Comput.-Hum. Interact.*, 7(4):461–476, December 2000.

[107] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, page 167–176, New York, NY, USA, 2009. Association for Computing Machinery.

[108] Arpita Ray Sarkar, G. Sanyal, and S. Majumder. Article: Hand gesture recognition systems: A survey. *International Journal of Computer Applications*, 71(15):25–37, June 2013. Full text available.

[109] Stefan Schneegass and Alexandra Voit. Gesturesleeve: Using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, ISWC '16, page 108–115, New York, NY, USA, 2016. Association for Computing Machinery.

[110] Teddy Seyed, Chris Burns, Mario Costa Sousa, Frank Maurer, and Anthony Tang. Eliciting usable gestures for multi-display environments. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, page 41–50, New York, NY, USA, 2012. Association for Computing Machinery.

[111] Sheng Shen, He Wang, and Romit 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, pages 85–96, New York, NY, USA, 2016. ACM.

[112] P. B. Shull, S. Jiang, Y. Zhu, and X. Zhu. Hand gesture recognition and finger angle estimation via wrist-worn modified barometric pressure sensing. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):724–732, 2019.

[113] Shaishav Siddhpuria, Sylvain Malacria, Mathieu Nancel, and Edward Lank. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, 2018. Association for Computing Machinery.

[114] S. Song, C. Hu, B. Li, X. Li, and M. Q. . Meng. An electromagnetic localization and orientation method based on rotating magnetic dipole. *IEEE Transactions on Magnetics*, 49(3):1274–1277, March 2013.

[115] S. Song, W. Qiao, B. Li, C. Hu, H. Ren, and M. Q. . Meng. An efficient magnetic tracking method using uniaxial sensing coil. *IEEE Transactions on Magnetics*, 50(1):1–7, Jan 2014.

[116] S. Song, H. Ren, and H. Yu. An improved magnetic tracking method using rotating uniaxial coil with sparse points and closed form analytic solution. *IEEE Sensors Journal*, 14(10):3585–3592, Oct 2014.

[117] Shuang Song, Chao Hu, Baopu Li, Xiaoxiao Li, and Max Q.-H. Meng. An electromagnetic localization and orientation method based on rotating magnetic dipole. *IEEE Transactions on Magnetics*, 49:1274–1277, 03 2013.

[118] Shuang Song, Hongliang Ren, and Haoyong Yu. An improved magnetic tracking method using rotating uniaxial coil with sparse points and closed form analytic solution. *IEEE Sensors Journal*, 14:3585–3592, 2014.

[119] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. 2017.

[120] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. Watchsense: On- and above-skin input sensing through a wearable depth sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3891–3902, New York, NY, USA, 2017. Association for Computing Machinery.

[121] Lee Stearns, Uran Oh, Leah Findlater, and Jon E. Froehlich. Touchcam: Realtime recognition of location-specific on-body gestures to support users with visual impairments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), jan 2018.

[122] Evan Strasnick, Jessica R. Cauchard, and James A. Landay. Brushtouch: Exploring an alternative tactile method for wearable haptics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3120–3125, New York, NY, USA, 2017. ACM.

[123] Evan Strasnick, Christian Holz, Eyal Ofek, Mike Sinclair, and Hrvoje Benko. Haptic Links: Bimanual haptics for virtual reality using variable stiffness actuation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 644. ACM, 2018.

[124] Paul Strohmeier, Roel Vertegaal, and Audrey Girouard. With a flick of the wrist: Stretch sensors as lightweight input for mobile devices. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, page 307–308, New York, NY, USA, 2012. Association for Computing Machinery.

[125] CyberGlove Systems. CyberGlove II. `http://www.cyberglovesystems.com/cyberglove-ii`. Accessed: 2019-05-09.

[126] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. *TelemetRing: A Batteryless and Wireless Ring-Shaped Keyboard Using Passive Inductive Telemetry*, page 1161–1168. Association for Computing Machinery, New York, NY, USA, 2020.

[127] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, and et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. Graph.*, 35(4), July 2016.

[128] Trung Duc Than, Gürsel Alici, Hao Zhou, and Weihua Li. A review of localization systems for robotic endoscopic capsules. *IEEE Transactions on Biomedical Engineering*, 59:2387–2399, 2012.

[129] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phuc Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, SenSys '18, pages 54–67, New York, NY, USA, 2018. ACM.

[130] Olivier AJ Van der Meijden and Marlies P Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical endoscopy*, 23(6):1180–1190, 2009.

[131] Radu-Daniel Vatavu. User-defined gestures for free-hand tv control. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroITV '12, page 45–48, New York, NY, USA, 2012. Association for Computing Machinery.

[132] D. M. Vogt and R. J. Wood. Wrist angle measurements using soft sensors. In *SENSORS, 2014 IEEE*, pages 1631–1634, Nov 2014.

[133] Cheng-Yao Wang, Min-Chieh Hsiu, Po-Tsung Chiu, Chiao-Hui Chang, Liwei Chan, Bing-Yu Chen, and Mike Y. Chen. Palmgesture: Using palms as gesture interfaces for eyes-free input. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, page 217–226, New York, NY, USA, 2015. Association for Computing Machinery.

[134] L. Wang, K. Sun, H. Dai, W. Wang, K. Huang, A. Liu, X. Wang, and Q. Gu. Witrace: Centimeter-level passive gesture tracking using ofdm signals. *IEEE Transactions on Mobile Computing*, pages 1–1, 2019.

[135] Robert Wang, Sylvain Paris, and Jovan Popoviundefined. 6d hands: Markerless hand-tracking for computer aided design. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 549–558, New York, NY, USA, 2011. Association for Computing Machinery.

[136] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3):63, 2009.

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

[138] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. Skinmarks: Enabling interactions on body landmarks using conformal skin electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3095–3105, New York, NY, USA, 2017. Association for Computing Machinery.

[139] Hongyi Wen, Julian Ramos Rojas, and Anind 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, page 3847–3851, New York, NY, USA, 2016. Association for Computing Machinery.

[140] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 86:1–86:12, New York, NY, USA, 2018. ACM.

[141] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):113, 2017.

[142] Eric Whitmire, Farshid Salemi Parizi, and Shwetak Patel. Aura: Inside-out electromagnetic controller tracking. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2019.

[143] Eric Whitmire, Laura Trutoiu, Robert Cavin, David Perek, Brian Scally, James Phillips, and Shwetak Patel. Eyecontact: scleral coil eye tracking for virtual reality. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 184–191. ACM, 2016.

[144] Mathias Wilhelm, Daniel Krakowczyk, and Sahin Albayrak. Perisense: Ring-based multi-finger gesture interaction utilizing capacitive proximity sensing. *Sensors*, 20(14), 2020.

[145] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M. Kitani. Back-hand-pose: 3d hand pose estimation for a wrist-worn camera via dorsum deformation network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 1147–1160, New York, NY, USA, 2020. Association for Computing Machinery.

[146] Haijun Xia, Tovi Grossman, and George Fitzmaurice. Nanostylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 447–456. ACM, 2015.

[147] Guofang Xiao, Ester Bonmati, Stephen Thompson, Joe Evans, John Hipwell, Daniil Nikitichev, Kurinchi Gurusamy, Sébastien Ourselin, David J Hawkes, Brian Davidson, et al. Electromagnetic tracking in image-guided laparoscopic surgery: Comparison with optical tracking and feasibility study of a combined laparoscope and laparoscopic ultrasound system. *Medical physics*, 45(11):5094–5104, 2018.

[148] Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. Lumiwatch: On-arm projected graphics and touch input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, 2018. Association for Computing Machinery.

[149] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D Wilson, and Hrvoje Benko. Mrtouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics*, 24(4):1653–1660, 2018.

[150] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[151] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Magic finger: Always-available input through finger instrumentation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 147–156, New York, NY, USA, 2012. Association for Computing Machinery.

[152] Hui-Shyong Yeo, Juyoung Lee, Hyung-il Kim, Aakar Gupta, Andrea Bianchi, Daniel Vogel, Hideki Koike, Woontack Woo, and Aaron Quigley. Wrist: Watch-ring interaction and sensing technique for wrist gestures and macro-micro pointing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[153] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. Opisthenar: Hand poses and finger tapping recognition by observing back of hand using embedded wrist camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 963–971, New York, NY, USA, 2019. Association for Computing Machinery.

[154] Sang Ho Yoon, Ke Huo, Vinh P. Nguyen, and Karthik Ramani. Timmi: Finger-worn textile input device with multimodal sensing in mobile interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 269–272, New York, NY, USA, 2015. ACM.

[155] Sang Ho Yoon, Ke Huo, and Karthik Ramani. Plex: Finger-worn textile sensor for mobile interaction during activities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 191–194, New York, NY, USA, 2014. ACM.

[156] Andre Zenner and Antonio Krüger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1285–1294, 2017.

[157] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T. Inan, Thad E. Starner, and Gregory D. Abowd. Tapskin: Recognizing on-skin input for smartwatches. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, page 13–22, New York, NY, USA, 2016. Association for Computing Machinery.

[158] Maotian Zhang, Qian Dai, Panlong Yang, Jie Xiong, Chang Tian, and Chaocan Xiang. Idial: Enabling a virtual dial plate on the hand back for around-device interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), mar 2018.

[159] Yang Zhang and Chris 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, pages 167–173, New York, NY, USA, 2015. ACM.

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

[161] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1491–1503, New York, NY, USA, 2016. Association for Computing Machinery.

[162] Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. Aurasense: Enabling expressive around-smartwatch interactions with electric field sensing. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 81–86, New York, NY, USA, 2016. Association for Computing Machinery.