Spatial Computing in Context

Ishan Chatterjee

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

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Shwetak Patel, Chair
Vikram Iyer
Steve Seitz

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Spatial Computing in Context

Ishan Chatterjee

Chair of the Supervisory Committee:
Professor Shwetak Patel
Computer Science & Engineering

The desktop computing paradigm has revolutionized the way we live, work, and communicate, but our interaction with these devices is limited by the physical location and boundaries of the devices themselves. More recently, mobile computing has extended these capabilities on the go, but the input and output mechanisms remain tied to the device surface, without interplay with the physicality of the surrounding environment. In contrast, spatial computing seeks to define a new generation of computing digital guidance, communication, and contextual information and derived from and integrated directly into the surrounding environment. This approach allows for *in-context* computing, where digital support is provided either actively or passively *while* users engage in other tasks. This shift introduces challenges in system interaction and input, which must operate seamlessly within dynamic contexts, but it also offers opportunities to harness spatialized information to enhance natural user interactions.

In this research, I explore spatial computing from three angles: *spatial computing interaction*, *spatial computing sensing*, and *spatial computing control*. From the perspective of spatial computing interaction, we investigate how electrical engineers can benefit from spatial augmentations while debugging printed circuit boards, and develop an AR workbench that tailored to their needs. For spatial computing sensing, we design earbuds that isolates the user's voice and filters out disruptive background noise by leveraging the spatial information of interfering audio sources. Finally, for spatial computing control, we design a ring peripheral that facilitates interaction across a variety of always-available surfaces, thus supporting mixed reality device input from on-the-go to high-precision scenarios.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Personal computing revolutionized our productivity and play, bringing instant access to information, communication, and digital creation into our homes and offices. Mobile phones unplugged the computer and put it in our pockets and purses, allowing us to take it into the world. Despite that, much of the interaction remains tied to the device itself, without reference to the user's surrounding environment. In these conventional setups, the focus is solely on the devices themselves, pulling users into their virtual worlds and out of the users' own surrounding ones. The spatial computing paradigm seeks to enable input and output to move beyond the boundaries of the physical device, sensing and augmenting aspects of the surrounding physical world. Rather than directing users' attention towards the device itself, this allows users to interact and perform tasks in their world (see Fig. 1.1), opening the door to computing acting as Weiser envisioned it: "a helpful, invisible servant".

In this proposal, I adopt Greenwold's original definition of *spatial computing* as "human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces"[1] [51]. This aspect of being rooted in the real-world orients spatial computing toward tasks that occur beyond the office desk. For example, the second-generation of Microsoft HoloLens and Magic Leap augmented reality (AR) devices were largely geared toward industrial applications such as manufacturing line guidance, surgical operations or in-field visualizations. Smartglasses like North Focals, Meta Rayban Glasses, or Oppo Glass 2 seek to support consumers during scenarios like walking navigation or visual querying. These types of tasks require spatial computing technologies to work in a given *context*, acknowledging

[1]While this definition subsumes the more recent uses of spatial computing in marketing material as a computer with a stereoscopic, head-mounted display, it is also more general: including any spatially-aware computing device, from a Nintendo WiiMote to headphones delivering spatial audio.

the user's current workflow or situation.

Dey and Abowd use *context* to refer to "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [36]. Although this definition is typically leveraged for defining *context-aware* computing – applications that are actively responsive to changes in context – in this proposal I instead focus on what I call *in-context* computing. In-context computing is used to assist the user in an application that exists beyond the computer itself, such as while they're performing another task or in a particular environment. For example, in-context computing could be an in-flight computer display meant to relay information to a pilot while flying or a conversational voice assistant for navigational queries as the user is walking. This would be in contrast to word processing on a laptop computer or a mobile game on a smartphone, where the application and user task are one and the same, and the application has no relation to the user's situation or environment. In-context computing can offer assistive extensions of the human's capability to allow for multitasking or greater efficiency in the task at hand.

As they are both tied to the environment and task-at-hand, spatial and in-context computing are two sides of the same coin. **I posit that by implicitly leveraging the *spatial* context of a user's environment, interactive technologies can better serve users as they operate within a situation, for example, while participating in another task or operating while mobile.** Through system building, I explore design constraints and considerations that follow from making spatial computing operate within a given situation, environment, or task. For instance, the latency, precision of interaction, form factor, robustness, power, social acceptability, comfort, and intrusiveness are a handful of key constraints that arise when designing an in-context computing system. To provide a comprehensive perspective, I approach this topic from three angles: *spatial computing interaction*, *spatial computing sensing*, and *spatial computing control*. These areas are highly interrelated, and although their boundaries are not sharply defined, they collectively form an end-to-end stack for spatial human-computer interaction (HCI). Spatial sensing captures

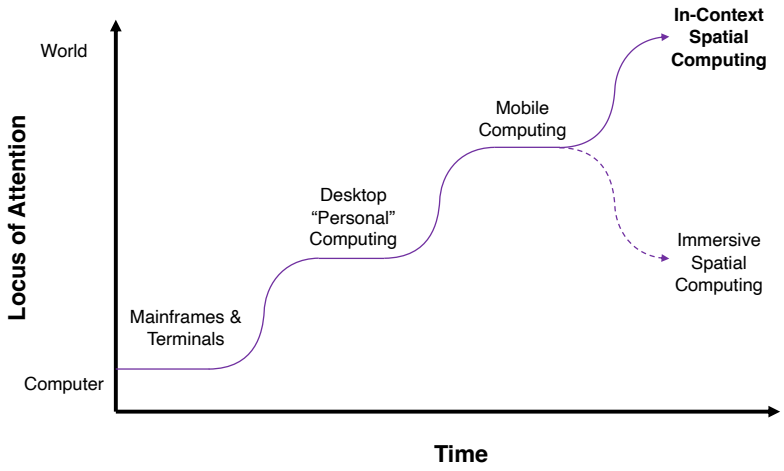


Figure 1.1: *As computing evolves, the user's locus of attention can now move from just tasks on the computer itself to concurrently supporting tasks in the real world. Users calculated a finanical model or authored documents on mainframes and desktops. Mobile computing has allowed for the user to take their computing into the world, such as controlling music on an iPod while running or taking a call in the car during a morning commute. However, interaction with, and an understanding of, the surrounding environment is still limited. This thesis focuses on the design and engineering of in-context spatial computing, which can offer unprecedented support to users during real-world tasks. Think smartglasses that answer queries about items in front of you during shopping or mixed reality headsets that augment surgical navigation guidance during operations. (In contrast, immersive spatial computing like VR has a virtual, computing-centric focus for tasks like gaming and collaboration.) As each paradigm has a different locus of attention and therefore associated tasks, each will continue to exist even as new ones arise. Graphic inspired by [56].*

signals from the world and the user, which can then be processed to enable spatial control of devices. This control mechanism is integrated with UI, UX design, and output, resulting in a complete spatial interaction. Throughout my studies, I have explored this thesis through various efforts (see Fig. 1.2). In this dissertation, I support this thesis by presenting a selection of these works.

Spatial computing interaction: Beginning with spatial computing interaction, the focus is on co-designing spatial input and output mechanisms that align with the user's workflow, enabling seamless support within their work context. Specifically, in Augmented Reality Debugging Workbench (Chapters 3 and 4), we ask: can augmented reality assist electrical engineers in their printed circuit board debugging workflows, and how can it best support them? (**RQ1**) To answer this question, we design and develop an augmented reality workbench for electrical engineers. To facilitate in-context computing, we co-design the interactions around their existing workflows, leveraging their workbench surface as a spatial canvas for projection and input.

Spatial computing sensing: Moving on to spatial computing sensing, the emphasis lies in understanding and utilizing the spatial characteristics of a given situation to enhance computing performance. Specifically, I focus on spatial audio capture sensing in noisy, real-world environments. In ClearBuds (Chapter 5), we ask: How can we effectively leverage spatial information, in addition to frequency information, about competing environmental sound sources to remove them and allow for clear calls? (**RQ2**). We develop a pair of wireless earbuds that uses multiple time-synchronized audio channels and a time-domain network to utilize the spatial distribution of the target speech source and interfering noise sources for speech enhancement.

Spatial computing control: Lastly, attention is given to the control of future spatial computing devices, especially for scenarios that extend beyond just the desk. From mice

to touchscreens, surface-bound interaction remains a cornerstone of ergonomic, precise, and subtle input methods. However, the requirement for an appropriate surface (e.g., trackpad, touchscreen) or handheld use (e.g., mouse) poses significant challenges for the evolving landscape of extended reality (XR) interaction. Future XR devices may operate across a variety of settings, including desktop-like experiences at a table (such as those found on Meta Quest Pro and Apple Vision Pro), on the go scenarios (like North Focals and Oppo Glass 2), and situations in between (e.g. consuming content on the couch). As spatial computing devices decouples the display from the interactive space, the world itself can become an always-available input surface. In FlowRing (Chapter 6), we tackle: How can we provide precise, ergonomic, and subtle surface-bound interactions to support spatial computing input across a variety of surfaces and scenarios? (**RQ3**). We present a ring-form-factor device with processing algorithms that enables interactions across a range of ad hoc surfaces – desks, pants, palms, and fingertips – and seamless switching between them.

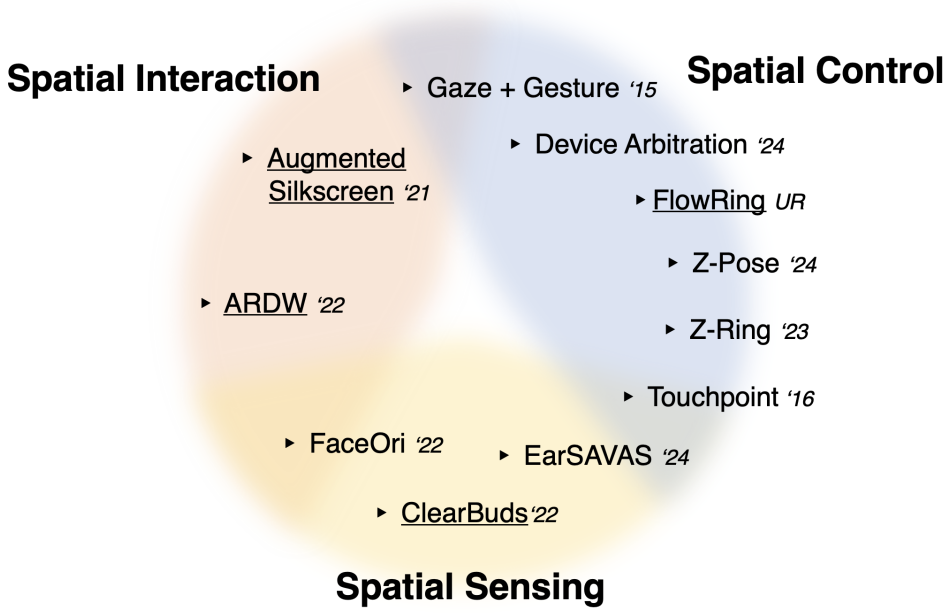


Figure 1.2: *During my studies, I have had the privilege of exploring spatial computing interaction, sensing, and control through a number of collaborative efforts. Many projects do not sit neatly in a single domain – instead, they are subject to design considerations from multiple perspectives – therefore, the blurred Venn diagram. For this dissertation document, the discussion focuses on the underlined works.*

Chapter 2
RELATED WORK

In this section, we review related work for each of the areas of focus: spatial computing interaction (Section 2.1), sensing (Section 2.2), and control (Section 2.3). We contextualize the work within the general thesis and then narrow to the specific contributions of the individual works that realize the hypothesis.

2.1 Spatial Computing Interaction: Augmented Task Guidance

2.1.1 Augmented Reality for Industrial Tasks

Our own senses are mediated by space – how we see, how we hear, and how we touch. Through our development, we generate natural intuitions about our body's relation to space [121]. Kirsh put forth the idea that the strategies employed to navigate our physical space play a crucial role in structuring our cognitive processes and actions [79]. Augmented reality (AR) capitalizes on spatial cognition and memory by establishing a meaningful association between information and physical objects/locations in the real world [13]. Therefore, tapping into these natural faculties can better lend itself to Weiser's vision of computing as "an extension of our unconscious" [166]. AR has long been seen as a paradigm that can decrease the barrier between virtual and physical information transfer [21]. This transfer process can consist of two components: the presentation of the information to the user, generally in the form of visualizations; and the ability for the user to interact with the visualization, perhaps enabling the user to query for additional information.

Prior work has shown that AR systems presenting spatially-tracked visualization, even with no interaction component, can be effective in reducing error rate and mental effort across industrial tasks. In 1992, Caudell and Mizell at Boeing built the "HUDset" [24] as a

demonstration toward four applications: building a wiring harness, plugging into a wiring connector, laying up composite cloth, and visualizing part disassembly. The system used a magnetic tracking system and an opaque, 7-degree field of view, monocular eyepiece. While the system relied on the user to fuse left and right eye images, had limited FoV, and suffered from significant lag, it was a bellwether for what could be possible with augmented reality (a term they coined along the way). Feiner et al.'s seminal KARMA system [40] similarly sought to provide augmented instruction in support of complex 3D tasks, for example, to guide the user through the maintenance of the laser printer with overlaid leader lines, callouts, and arrows bound to different components of the machine. Schwerdtfeger and Klinker [139] investigated the use of AR for order picking, designing a flight tunnel visualization to guide users to grab the part from the correct bin, and resulting in zero errors during all 1512 picks. Tang et al. [154] compared the effectiveness of different instructional media in an assembly task, comparing print media, heads-up instructions, and spatially-locked AR instructions. The AR system displayed task information as 3D objects in the user's field of view, aligning them with the workspace. The results revealed an 82% reduction in the error rate when overlaying 3D instructions on actual work pieces, particularly in minimizing cumulative errors. Additionally, the AR condition showed a decrease in mental effort, suggesting the potential of AR to alleviate cognitive demands on the user while in context. Rosenthal et al. [133] augmented a laptop screen with an attached microprojector. They investigated a variety of tasks, such as evaluating a breadboard for correctness and cutting wires to length, finding a 27% overall speed-up and 32% reduction in errors with projections compared to just the screen-only.

Extending AR experiences to include interaction makes them even more powerful. Such interactions might enable actions such as the selection of elements in the physical world to be used as part of a virtual tool operation. For example, Digitaldesk [167] demonstrated such interactions with examples such as allowing the user to move a number from a physical price list into a virtual calculator. In this first section on spatial computing interaction, our work continues the line of research into helpful augmented reality in industrial contexts. We

explore the design space of both AR visualization and AR interaction, to understand how they might enhance electrical engineers' workflow with PCBs. We observe that their workflow requires frequent context switching between various virtual design files and spatial navigation on their physical design. To provide background on these current workflows, we now take a slight detour from spatial computing technologies to examine their current tools.

2.1.2 Limitations of Current Tools

During the process of debugging a new PCB design, engineers must constantly move between their schematic, layout, and physical representations in order to validate their design or understand the nature of a design failure.[1] Through current ECAD tools, this can be a time-consuming and error-prone process which typically involves manually following correspondences and nomenclatures across different software applications, often using each application's textual "find" command. To locate the corresponding area-of-interest on the board, the engineer must textually match the reference designator against the board's silkscreen (if printed) or visually pattern match layout representation to the physical PCB, which can become more challenging with dense components or different orientations. As a debugging procedure typically involves tens to hundreds of these correspondences, this can quickly become tedious with the ever-increasing complexity of board designs only exacerbating the challenge. In our work, we interview electrical engineers on their workflow to get a better picture of pain points in their current processes. We design our spatial computing system with an eye towards relieving these pain points.

---

[1]Some helpful technical background and terminology: The electrical engineering design process typically starts with designing a circuit to meet a set of functional requirements. Using an electronic computer-aided design (ECAD) tool, the logic of the circuit is formalized via schematic capture into a *schematic diagram*, which visualizes the circuit's components as symbols and the circuit's interconnections (*nets*) as topological lines between the components' pins. This logic is then transferred to a *layout diagram*, where components and connections are placed in a physical coordinate space. Finally, the design is then physically fabricated and assembled into a functional PCB, where the components are soldered onto the surface of a fiberglass board with conductive pads, vias, and traces running buried within its layers. For images, see online tutorials on Sparkfun (`https://learn.sparkfun.com/tutorials/pcb-basics/all`) or Adafruit (`https://learn.adafruit.com/making-pcbs-with-oshpark-and-eagle`).

2.1.3 Augmented Reality for PCBs

While PCBs are often considered the staple of industry level electronics, breadboards are often used by students and hobbyists for their solderless reconfigurability that enables rapid iteration. However, they are rarely used as part of the hardware development process in industry. While recent work in the HCI community aimed at the student population has demonstrated a number of breadboard augmentation techniques [114, 38, 176, 77], PCBs are substantially more intricate, requiring much more careful augmentation. In this section, we discuss more directly comparable related work in the realm of specifically augmenting PCBs.

Visualization Tools A few tools support visualizing certain component metadata, such as location, directly on the PCB. InspectAR [61] is a recently released tool that uses mobile AR to overlay elements of the layout and associated metadata onto a camera view of the PCB displayed on a mobile tablet or PC. It is targeted toward supporting industry professionals, with couplings to industry standard ECAD tools. The tool does not seem to support direct interaction with the PCB itself, measurement interactions, or a topological schematic view. The sales webpage offers strong testimonials speaking to the increased assembly and debugging efficiency from decreased context-switching, claiming "an average 30% reduction in lab-time." While these indications speak strongly to the hypothesis that mixed reality visualization of layout metadata on PCB can increase efficiency, a systematic study is yet to be published. The Mascot [131], a robotic workbench from Robotas, helps to support operators performing hand assembly of through hole components by steering a projected laser spot to the installation location on an anchored PCB. Similarly, Hahn et al. [54] generated an AR tool with textual and graphical cues delivered through a smartglass for assisting workers performing PCB assembly, indicating that the tool allowed for errorless part picking and assembly. Hahn et al.'s tool, InspectAR and Mascot all provide board-locked augmented instruction for PCB workflow, driving information from the virtual design files to the user's view of the PCB. Our work broadens the design space seeking to also incorporate augmented interaction and measurement to pass data in the opposite direction, that is, interactive capture in the PCB

view can be passed to the virtual design files.

Adding Interactivity and Measurement Pinpoint [146] is a tool designed to assist in PCB debugging by allowing users to modify and measure the circuit *in situ* after the PCB is fabricated. The tool modifies the layout of a PCB by inserting breakable connections on some traces. While not using augmented reality per se, the tool connects the virtual and the physical by using GUI-controlled relays to make and break these connections. For form factor designs and mass-produced PCBs, modifying the layout for test is typically restrained to adding test points on critical nets for bed-of-nails, on-line testing or manual access for workbench debugging. Our work seeks to support existing debugging workflows that do not modify the PCB design, and instead ease access to measurement points by guiding users with augmentations.

More relevant to our work, BoardLab presents a magnetically tracked stylus that enables interactions from board to schematic, such as selecting and identifying components on the schematic by touching the components on the board as well as taking voltage measurements and having the measurement annotated on the schematic [49]. Although the system looks promising, no formal evaluation was reported. Our work studies whether the interactions afforded by such a stylus would be helpful to electrical engineers as they actively debug, as well as exploring interactions that are synergistically enabled as augmented interaction and measurement is paired with simultaneous augmented visualization.

2.2 Spatial Computing Sensing: Spatial Audio Capture

2.2.1 Sensing Space and Distance

Historically, systems have used a wide variety of sensing methodologies to understand space. Radar, sonar, and, more recently, GPS were all initially developed for wartime applications with battlefield-scale positioning. Within the room-scale, Sutherland's pioneering augmented reality system explored the use of multifrequency continuous-wave ultrasound, though the first system relied on a set of taut lines on reels sensed by positional encoders [150]. The

calculated device pose was used to update the see-through graphics in real-time. Modern AR and VR head-mounted devices utilize a passive method of 6DOF positioning, derived from simultaneous localization and mapping (SLAM) robotics [80]. Images from multiple on-headset cameras are processed to generate a 3D visual map of world keypoints. Intermediate positions are calculated by integration of a calibrated inertial measurement unit (IMU) and are fused with vision via an Extended Kalman Filter. This technique has been key to allow for these HMDs to be mobile and used in-the-field without need for external tracking hardware. For dense mapping of the geometry of the surrounding environment, a number of headsets [106, 100, 159] employ a time-of-flight depth camera which can be leveraged for semantic understanding of the surrounding geometry, accurate interactive plane detection, and hand tracking.

With the exception of SLAM, each of these systems utilizes precise timings of reflected or transmitted waves with a known velocity to calculate distance. This same technique is utilized in the audio domain to localize sound sources. Our ears are separated by a known baseline. Audio from an off-center source arrives to our ears at slightly different times resulting in an interaural time difference (ITD) given the speed of sound [99]. This phenomenon along with interaural level differences (ILD) and frequency modulations resulting from the head-related transfer function are the primary cues that allow humans and other animals to precisely localize sound sources.

In noisy environments, such as on a busy street, multiple competing sound sources layer on top of one another, creating a noisy mixture. Dubbed the cocktail party effect, our brain's ability to focus auditory attention on a given sound source while filtering out other stimuli has been studied since at least the early 1950s [122]. Interaural time differences as well as the audio frequency content both play a role in source separation [58]. Here we focus on producing a system that can utilize both cues together to enhance a user's voice from amongst a noisy background. We apply this technique specifically to a set of wireless earbuds. Because of their portability, these systems are often used to take calls in noisy, dynamic environments. Real-time speech enhancement is a long-standing problem in the

signal processing and machine learning communities. While recent advances in the machine learning community have shown promising results, none of them have been demonstrated on wireless earbuds. Furthermore, the vast majority cannot run on mobile devices and meet these real-time constraints. As a result, endfire beamforming configurations remain popular on most consumer mobile phones and earbuds [135, 6, 140, 63]. Below, we briefly discuss beamforming, single channel speech enhancement networks, and binaural networks. By creating a wireless acoustic network between two earbuds and a novel light-weight hybrid time-domain and spectrogram-based neural network, we show for the first time that real-time two-channel neural networks can outperform current real-time speech enhancement approaches for wireless earbuds.

2.2.2 Spatial Audio Capture

Beamforming techniques A common approach to enhancing speech is to design a beamforming microphone array to be more sensitive to sounds coming from the direction of the user's mouth [158] or voice [1]. Since signal-processing based beamforming is computationally lightweight compared to other speech enhancement techniques, these techniques are deployed on many commercially available audio products today such as smart speakers [4], mobile phones [135], and earbud devices like Apple AirPods [6]. However, the performance of beamforming is limited by the geometry of the microphones and the distance between them [158, 63]. The form factor of devices like AirPods restricts both the number of microphones on a single earbud and the available distance between them, limiting the gain of the beamformer. While beamforming simultaneously across two earbuds could provide better performance in principle, current wireless architectures are limited to streaming from a single earbud at a time [14]. Furthermore, adaptive beamformers such as MVDR [41], while showing promise with relatively few interfering sources, are sensitive to sensor placement tolerance and steering [187, 18]. Finally, beamforming leverages spatial cues only and does not use acoustic cues and perceptual differences to discriminate sources, information that machine learning methods leverage successfully.

Single-channel speech enhancement Recent deep learning techniques have led to significant progress in single-channel speech enhancement methods. These models typically operate on spectrograms to separate the human voice from background noise [178, 109, 39, 113, 169, 42, 143]. However, recent trends have opted to operate directly on time domain signals [96, 44, 118, 35, 98], yielding performance improvements over spectrogram approaches. Commercially available noise suppression software such as Krisp [84] and Google Meet [47] have successfully deployed single-channel models in real-time and are available for use on mobile phones and desktop computers; processing is performed on the cloud. However, single-channel models cannot effectively capture the spatial information and hence fail to isolate the intended speaker when there are multiple speakers present.

Multi-channel source separation and speech enhancement Multi-channel methods have been shown to perform better than their single-channel source separation counterparts [180, 32, 187, 52, 155, 66]. Binaural methods have also been used for source separation [149, 55, 89, 127] and localization [157, 97, 81]. Our method improves existing binaural methods by combining a time-domain neural network with spectrogram-based frequency masking networks and optimizing them to enable real-time processing on a phone. Recent works [152, 153] use multiple microphones on a smartphone for speech-enhancement; however, neither of them demonstrates real-time performance of a smartphone. In contrast, we demonstrate the first system that achieves real-time speech enhancement using microphones on the two wireless earbuds. Furthermore, since the distance between the earbuds is larger than the distance between microphones on a typical mobile phone, we can attain a better baseline than a mobile phone implementation while also retaining the ability to speak hands-free.

2.3 Spatial Computing Control: Surface Interaction Anywhere

To deliver spatial computing experiences, a head-mounted device (HMD) form factor can provide a number of benefits: (1) the potential to deliver display content overlaid directly on the person's field of view (FOV), (2) the potential to scale that visual area from something

smaller, such as an information display, to something much larger, like an immersive view, and (3) the potential of that display, especially in the case of an informational display, to be mobile and "always available" as a wearable. This has resulted in a huge amount of industry and academic investment, accelerating within the last decade, toward engineering head mounted devices across all flavors of Milgram's extended reality (XR) spectrum [107]. Smartglasses can operate as a head-up display for timely and spatially relevant digital content. Higher-end mixed reality devices can maintain spatial coherence of the augmentations during movement, providing the illusion of world-locked content. Virtual reality devices can provide a sense of immersion that transports the user to a new environment. However, now that the output display space exists projected into the world, input is more challenging. Unlike the trackpads and touchscreens and keyboards of previous computing paradigms, input may no longer exist tied to the device surface. This is especially challenging for future XR devices which may work across the spectrum, from immersive, productivity tasks to quick, mobile scenarios. In Chapter 6, we propose an input device to enable seamless control between stationary and on-the-go interaction and across both large and small interaction surfaces, all with the same set of hardware and sensors. To contextualize and motivate our design, we discuss current methods of XR device control in market, then focus on the space of interaction devices—both wearable and environmental—that span large and small interaction surfaces.

2.3.1 Current Methods of Spatial Computing Control

To control immersive devices, 6DOF controllers are typically employed. These controllers use camera-based tracking in two flavors: outside-in or inside-out tracking. Outside-in methods involve a set of cameras on the headset tracking a constellation of infrared LEDs on the controller. Two limitations are that the controllers must be within the headset camera's FoV and have a large ring of LEDs positioned in a way that the constellation is not obstructed. Inside-out tracking leverages multiple cameras within the controller with the same SLAM algorithm discussed in the previous section. With the additional compute and cameras, this technique tends to be higher power and require a few seconds upon start

up for map relocalization [67]. The controllers offer compelling experiences for high-fidelity, immersive gaming, where low-latency hand positioning and the gamepad controls can provide precise manipulation and action interactions. However, as we think toward in-context spatial computing where users are performing another task like maintenance, surgery, or typing, it is essential to keep the hands unencumbered.

Indeed, augmented and mixed reality devices also offer articulated hand tracking using computer vision on depth camera or stereo camera data [106, 104, 156]. This allows for direct manipulation of virtual buttons and content, similar to the interaction techniques studied with data gloves in the early days of VR [190]. While hand tracking now does not require donning a glove, it does require the user to keep their hands within the field-of-view of the camera. Therefore, this method of control, as well as those that involve keeping a controller within camera FoV, suffer from "gorilla arm syndrome" [17]. As we consider augmented reality that will be used in more public settings with lighter weight headsets, such as smartglasses, hand tracking-based interaction which uses large motions which can draw unwanted attention to the user posing social acceptability challenges, such as in meetings, walking down the street, or on a subway [129]. Furthermore, the privacy and power implication of having an always-on, on-headset cameras will also need to be evaluated in public and private scenarios.

To reduce fatigue, in previous work, I proposed and explored fusing eye gaze and hand gesture for input [31]. These two input streams are complementary with eye gaze providing low effort targeting, and gesture allowing for robust and intentful pinch and manipulation. The method can allow for high throughput control with low physical demand, however requires precise and robust eye tracking across people, headset motion, and re-wear. As sensing technology has matured, the mixed reality Apple Vision Pro [7] headset recently shipped this interaction model, using two eye tracking cameras and 17 glint LEDs per eye and six hand-tracking cameras plus illuminators. While this input method has the potential to scale to function across the spectrum of XR devices, breakthroughs in sensing complexity will be necessary to be realized in lighter weight devices. Additionally, more work is required to understand the social acceptability of eye tracking controlled devices in public

settings, especially from a bystanders standpoint. Similar to hand-only systems, current implementation still require gestures to be performed within headset cameras field-of-view.

As we consider the limitations of the above methods and the needs of future XR headsets to function across immersive and mobile spatial computing, a set of design constraints arise. First, for immersive scenarios, the input must be be precise, like hand tracking, but also ergonomic, like eye tracking. Second, for mobile scenarios, the interaction must be suitably discreet, accessible, and quick. Third, the input needs to be sensed without requiring optical line of sight from the headset to reduce sensing complexity for lighter weight devices and improve subtlety of interactions. Taken together, we propose a ring-based peripheral we call FlowRing that enables surface-supported interaction across a set of common surfaces. For precise tasks, users may use large surfaces such as a desk top or their leg for cursor control similar to a mouse. For on-the-go scenarios, the same peripheral can enable subtle microgestures, such as swipe and taps on their fingertip surface. FlowRing is designed for support all of the above interactions, as well as identify surfaces and switch interaction methods seamlessly. In the next sections, we explore large scale on-surface input and wearable microgesture systems.

2.3.2 Large Scale On-Surface Input

Surface-bound interaction remains a cornerstone of ergonomic, precise, and subtle input methods. For the desktop and mobile computing paradigms, mice, trackpads, and touchscreens have become the standard input devices that have withstood the test of time. To allow for world scale surface input, researchers have deployed a variety of on-surface sensor modalities to enable surfaces to identify user interaction across varying scales. Wall++ and GravitySpace demonstrate whole room scale surface input by instrumenting environmental surfaces (such as walls and floors) with capacitive and optical sensors, respectively [188, 19]. SAWSense [65], Scratch Input [57], and SurfaceIO [37] enable smaller scale surface interaction on the scale of furniture and device surfaces through mounted contact microphones and piezo vibration sensors. SurfaceIO [37] demonstrates the use of on-surface interaction on a mobile VR

headset device through the use of 3D printable touch surfaces; while allowing for mobile interaction on smaller surfaces, this approach is still confined to instrumented surfaces, inhibiting cross-surface switching.

2.3.3 Wearable Devices for Surface Interaction

Instrumenting the user offers both mobility to provide input across environments and the potential for cross-surface input from the same sensing system. However, many systems that do so use power-hungry active sensing techniques and demonstrate only a subset of the surface scales and mobility spectrum that FlowRing affords. On the end of the spectrum closer to FlowRing's tabletop interaction is Acustico [45], which allows on-surface interaction from a wrist-worn form factor; unlike FlowRing, it requires constant contact with the surface to perform interaction and works only with rigid surfaces that conduct acoustic signals. Z-Ring [162] achieves 2D tracking and contact detection, however only on asymmetric, precalibrated conductive surfaces, limiting utility. Approaching the on-body surface interaction of FlowRing are VibAware [76], 3DTouch [112], Ready Set Touch! [141], Magic Finger [179], DualRing [91], and LightRing [72], which all achieve surface interaction without signal transmission through the surface itself and thus are either demonstrated or likely to work across a wider array of surfaces (at the expense of some expressiveness). Both 3DTouch and Magic Finger mount optical flow sensors to the fingerpad itself, impeding practicality for an always available input device. Most similar to our work is LightRing, a ring which specifically offers continuous 2D control using a gyroscope and an optical sensor but does not enable the ability to lift the hand off the surface to clutch a cursor or tap on the surface to select. FlowRing addresses this aspect by accurately determining surface-bound contact, therefore can acting like a trackpad for 2D interaction beyond just swipe motions.

2.3.4 Adding Microgesutres

Ashbrook [10] defines microinteractions as device interaction under four seconds, citing Oulasvirta's investigation into user attention during mobile situations [116]. While engaging

in activities like navigation, eating and conversing, participants shifted their attention for four to eight seconds to the device before returning to the primary task. This means the friction to providing input while mobile must be even less, prompting Ashbrook to explore touch gestures and bezel manipulation on wristband-based devices [9, 8]. In this work, we similarly propose a wearable sensor to enable always-available microgestures that can be performed quickly. Rather than on-device bimanual interactions, we instead focus on single-handed microgestures between the thumb and index fingers.

The fingertips feature both a high spatial resolution of tactile sensing and highly dexterous control. Neurophysiological findings indicate that fingers have some of the largest representations in both the somatosensory and motor cortices as function of their size [138]. Zhai et al. demonstrated that leveraging small muscle groups like fingers demonstrated manipulation performance that exceeds input relying just on wrist, arm, and shoulder [181]. Amongst the fingers, user elicitation studies prompting users to suggest hand gestures for common digital functions resulted in most gestures being executed between the thumb and index fingers [25]. This tracks with anatomical analyses indicating the greatest dexterity is between these two fingers [174, 90, 86]. Similarly, DigitSpace compared comfort for thumb gestures across different fingerpad areas and found the tip of the index fingerpad to be the highest rated for both taps and swipe gestures [60]. Oh et al. found that the proprioception between thumb and index finger allowed for a better understanding of the moment of touchdown between the two fingers, enabling higher time domain accuracy in selection than controller or non-pinch microgestures [115].

Many systems have explored sensors mounted on the wrist, hand, and/or fingers to sense the more subtle motion of these appendages without limitations on headset field of view. These systems span a wide range of modalities: optical ([27], [74], [128], [101], [26], [94]), ([46], [186], [163]), bio-acoustic ([5], [111], [185]), ultrasonic ([64], [183]), mechanical ([151], [92], [85]), electromyography ([136]), and inertial ([53], [102], [43], [91]). The capability of these devices ranges from detecting pinches to recognizing discrete gesture sets to driving full kinematic models of the hand. A shared design goal of each of these systems is to keep the

hand itself unencumbered allowing for the user's primary task to be minimally affected, so called free-hand or device-free gesture systems. Amongst systems that enable thumb-to-index microgestures, AO-Finger [177] is most similar to FlowRing's multi-sensor approach. It leverages a combination of acoustic and optical signals to infer such microgestures, but it does not demonstrate the aforementioned on-surface interaction, such as 2D tracking. Most similar in affordance to FlowRing is Ready Set Touch! [141], which presents a nail-mounted IMU for classifying gestures performed on a surface and in air but does not yet discriminate between surfaces or achieve both 2D tracking and microgestures. Furthermore, the mounting location must be moved from index nail to thumb nail for surface versus in air gestures. FlowRing is designed to merge the affordances across the spectrum of interaction styles, all from the same device worn in an ergonomic location at the base of the index finger.

Table 2.1: *A table comparing the most relevant prior work to FlowRing. To our knowledge, FlowRing is the first ring device that enables both continuous on-surface interaction and microgestures as well as the first device to facilitate seamless transitions across these input techniques and across surfaces. It also does so while adopting an ergonomic location on the base on the index finger.*

Prior Work	**Sensing Modality**	**Form Factor**	**In-Air Microgestures**	**Continuous, On-Surface Tracking**	**Surface Context-Awareness**
ElectroRing [73]	Bio-impedance	✓ Index ring	? Pinch only	X	X
Z-Ring [162]	Bio-impedance	✓ Index ring	✓	? Conductive surfaces only	X
AOFinger [177]	Acoustic, IR reflection	✓ Wristband	✓	X	X
DualRing [91]	Bio-impedance, inertial	X Thumb + index ring	✓	X Surface tap only	X
LightRing [72]	Inertial, IR reflection	✓ Index ring	X	✓ But no tap	X
Acustico [45]	Acoustic	✓ Wristband	X	X Tap on rigid surface only	X
3DTouch [112]	Optical flow, inertial	X Finger pad	X	✓	X
VibAware [76]	Active acoustic	X Thumb + wrist	✓	X Gestures possible	✓
Magic Finger [179]	Optical flow, camera	X Finger pad	? Only pinch shown	✓	✓
Ready Set Touch! [141]	Inertial	X Nail-mounted	✓ Thumb nail mount	✓ Index nail mount	X
FlowRing (this work)	**Optical flow, inertial, acoustic**	✓ **Index Ring**	✓	✓	✓

Chapter 3

DESIGNING THE AR DEBUGGING WORKBENCH

In this chapter, we interview electrical engineers, investigating how spatial computing can support their PCB debugging workflow. To guide our interviews, we develop and leverage an interactive interface serving as a mock PCB called Augmented Silkscreen. These design learnings are then applied to the final system we build, ARDW, which is described in Chapter 4. This chapter was published in ACM DIS 2021 [28]. An illustrative video description can be found here: `https://youtu.be/TDSsPAMvjxo`.

3.1 Introduction

By 2030, the number of smart devices in the world is projected to reach 50 billion [103]. The proliferation of these devices can largely be attributed to the increasingly integrated nature of electronics and silicon, as per Moore's law. Just as the number of transistors in an integrated circuit (IC) have increased exponentially, so too have printed circuit boards (PCBs) become increasingly dense with electronic components. These denser and more complex PCBs pose greater challenges for electrical engineers during the debugging process. However, the tools used to support these engineers in debugging faulty PCBs during design and development remain largely unchanged. During the process of debugging a new PCB design, electrical engineers must constantly move between circuit diagrams, board layout diagrams, and the physical circuit board itself in order to validate their design or understand the nature of a design failure. The design and the layout might be distributed across both physical (i.e. paper) and virtual (i.e. software tool) mediums as well. The constant context switching, as well as manually looking for corresponding components across the different representations of the circuit, lends this process to be extremely time-consuming and error-prone, such that the

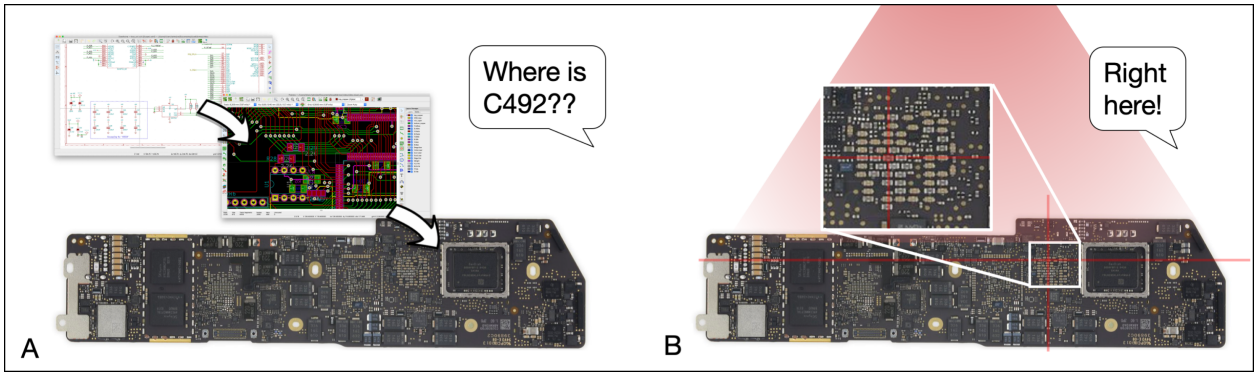


Figure 3.1: *(A) PCBs, such as the motherboard above, can contain thousands of components, most smaller than a grain of rice. Among other metadata, each component has a location, orientation, reference designator, and set of pins. Pins are connected via metallic traces buried in the board called nets. Basic tasks involved in debugging PCBs (such as finding a given component, pin, or net) typically involve flipping through multiple software files, such as the schematic ((A), top) and layout ((A), middle). (B) Augmented Silkscreen explores augmented reality interaction techniques to make this and other PCB debugging workflows more seamless and efficient.*

smallest optimizations to this process can have significant compounding benefits.

Augmented reality (AR) has been cited as an effective paradigm for reducing the overhead of tasks with repeated context switching, particularly those with spatial associations and affordances [12, 40, 24, 154, 110]. While there has been some amount of prior work exploring ways of using AR for debugging breadboards and PCBs, the primary focus has been on supporting hobbyist makers and students, in particular taking an educational perspective [38, 176, 145, 77]. In this work, we conduct a design exploration, investigating how this paradigm can be effectively applied to support the existing PCB debugging workflows of industry professionals through a series of needs finding and evaluative user studies. We design a set of AR interactions to enhance workflow productivity, and evaluate their utility via feedback interviews, illustrative video sketches, and remote simulation of certain PCB tasks. The scope of our work does not include the complete implementation of an AR tool, instead focusing our exploration on understanding user needs and designing AR interactions agnostic to AR implementation (head mounted device, projective AR, camera pass-through AR, etc.).

The goal of this work is to highlight and demonstrate to the HCI community the design considerations and research opportunities in this space.

The main contributions of this work include:

1. An initial, formative study identifying challenges in PCB assembly, bring-up, and debugging workflows to inform interaction design.

2. A set of augmented reality interaction techniques supporting workflows related to localization, annotation, and measurement operations of components, pins, and nets across the design files and the physical PCB.

3. A user feedback evaluation (n=6) for:

   (a) assessing the interaction techniques for user preference, usage, and likelihood of adoption, and

   (b) evaluating completion time and user confidence in component identification tasks.

3.2 Study 1: Formative Needs Finding

To gain an understanding of the needs of electrical engineers during their debugging workflows and characterize their existing workflows, tools and methods, we conducted a formative needs finding survey and semi-structured interviews.

3.2.1 Participants and Procedure

We recruited 8 participants who hold electrical engineering roles in academic labs and industry (high technology, consumer electronics firms). While all of our participants regularly design and debug their own PCB designs, their experiences spanned one-off or low-volume designs for research or hobby purposes, complex development boards with thousands of components, and mass-produced form factor logic boards shipping hundreds of thousands of units (see Table 3.1).

We conducted remote semi-structured interviews with the participants. Each of the interviews lasted for about 1 hour and consisted of 3 main parts:

1. We asked the participants about their current debugging flow, strategies, common pain-points, and needs.

2. We solicited feedback on initial speculative design concepts that we described using sketches and hypothetical use-cases. Participant were also invited to share any functionality or interactions they were missing in the currently existing tools.

3. We asked them whether collaboration was important in their day-to-day work. When relevant, we specifically asked about how the information is transferred when more than one person is involved in the workflow.

We supplemented the interview data with our own professional experience debugging PCB in both industry and academic institutions. We analyzed their responses via thematic analysis

Table 3.1: *Recruited participant backgrounds.*

	Field	**Experience**	**Primary Tool**	**Designs**	**Study 1**	**Study 2**
P1	Academia	Design, Release, Assembly, Functional Check, Rework	EAGLE	Mixed Signal, Embedded Systems, Wearables, Typically small, two-layer boards	X	X
P2	Industry	Design, Release, Engineering validation, Mass production, Field failure analysis	Cadence OrCAD/ Allegro/ PCB, Zuken CADSTAR	Mixed signal and high speed development boards (large format, thousands of components, 14 layer), form factor for wearable devices (12 layer)	X	X
P3	Academia	Design, Release, Functional check	EasyEDA	Antenna Patterning	X	
P4	Academia	Design, Release, Assembly, Functional check	Altium	High wattage power circuits	X	
P5	Industry, Hobby	Design, Release, Engineering validation, Mass production	Cadence Allegro/PCB	LED display, GPS radio module, Charging and battery protection circuits, FPC	X	X
P6	Industry	Design, Release, Engineering validation, Mass production	Cadence Allegro/PCB, Altium	Mixed signal, Actuator drivers, High-voltage designs, FPGA boards, form factor for wearables (4 layer), FPC	X	X
P7	Industry	Design, Release, Assembly, Engineering validation, Mass production, Field failure analysis	Cadence Allegro/PCB, KiCAD	Small form factor for wearables (12 layer), large form factor boards for gaming console (12 layer), FPC, RFPC	X	X
P8	Academia, Hobby	Design, Release, Engineering validation	Eagle	High-voltage designs, Aerospace, Power electronics	X	X

[20], first transcribing the interviews, then coding recurring themes, and finally noting outliers from the norm.

3.2.2 Findings

From our participants' responses in the interviews, we extracted four sub-tasks that constitute a framework for localizing errors during debugging:

1. Perform a visual inspection, measure output one sub-section at a time, and compare to expected values

2. Identify an anomalous measurement and hypothesize fault causes, such as defects in design or processing

3. Examine potentially contributing elements and make localized measurements to test hypotheses

4. Compare real measurements to expected values derived from schematics, layouts and datasheets

Most of our participants alluded to the challenges of context switching and information logging while debugging a PCB. They raised concerns about referencing a large set of information sources during debugging (on their computer monitor or sometimes printed paper: schematics, layouts, datasheets, bills of materials, emails; on their workbench: instrument measurements, PCBs) and the frequency with which they moved between these items: *"Very often, maybe multiple times per minute"* (P1, Quote Q1). *"I would say almost constantly until I get to fab C or D"* (P2, Q2). *"I would say at least multiple times a minute I'd switch back and forth."* (P7, Q3). *"Gotta go back and forth and each time you go back and forth you add more information to your schematic, and eventually find a value that doesn't line up... Probably five or six times a minute"* (P8, Q4).

Participants stated the information they cross-referenced most often in this process included component reference designators (toward the task of component localization), component values, pin or net locations (for the purpose of determining where to probe), net connectivity, measured values from instruments, IC pin assignments, and IC orientations.

The challenge of component localization was not shared by one participant who pointed out that, in doing the end-to-end PCB design process, she was able to memorize all of the components of the design. In addition, due to the high voltage nature of her work, her circuits generally included fewer but larger components than the other participants typically worked with, allowing her to include reference information on the silkscreen of her fabricated PCB. *"I made the PCB. I verified that my design works in a PCM simulation; I mean I never had a situation where I couldn't find my component [from memory]"* (P4, Q5). Another participant also expressed a similar sentiment *"I have it memorized"* (P5, Q6). Both noted that confirming orientation and pin assignments, as well as localizing small components on complex boards still pose a challenge for localization. Participants expressed interest in having component, pin, or net metadata within their view of their board, but looked to avoid interference: *"Yeah, that would be really helpful as long as it didn't interfere with my ability to see the PCB."* (P8, Q7). *"I usually like having two scenes. Like sometimes I don't want the information... just want to know the reference designator... but then sometimes if I'm debugging, like show me that info that I need: [lists various metadata categories]."* (P5, Q8).

Finally, participants cited other explicit processes where they felt their software and design tools fell short:

Assembly: While not all users assembled their own boards, those that did expressed frustration in matching the ordered components to the correct location and orientation on the PCB.

> "So you have to have a separate BOM that you make yourself, like an Excel document or a Google document. I'd have that, the schematic, and I have the layout up on my laptop so I'd be switching back and forth between all of them, trying to figure out where each component went. it was not fun." (P7, Q9)

Measurement: The need to localize probe points, trace net connections, and log measurements were shared as common pain points.

> "It's typically just like you look down at the board you make a measurement, you

> know, you might have a Google Doc with your testing records in it or something that you're documenting as you go." (P8, Q10)

Bring-up: Before green-lighting an entire production run, the bring-up process is followed after completing the first board assembly: 1. visual inspection, 2. confirm correct component stuffing, 3. confirm correct pin one locations (an indicator for verifying orientation), 4. perform open/short test on all voltage rails, 5. apply power with current-limited supply, 6. check each voltage rails for correct level, 7. perform functional sub-system checks. Some users resorted to general purpose software to prepare customized views ahead of time.

> "What I'll do is [that] I'll have a premade PowerPoint deck, and I have everything I need with all the steps and all the images I need, tables that I can fill in. So I'll, you know, identify all the pin one locations and what's stuffed and not stuffed with the picture that I make ahead of time." (P7, Q11)

Collaboration: A more frequent occurrence as a result of the recent COVID-19 pandemic, a handful of participants felt that working with collaborators who were less familiar with their own design, tackling a new developer kit, or approaching someone else's design posed new challenges.

> "I'm actually going to pass this design to this other engineer who's going to get some of those boards, who's not familiar with the design and I think for someone that's, you know, not familiar with the design [who] is trying to do like bring up, that would be extremely helpful to go or like look at a part or touch part with a stylus and have it pull its datasheet and point it to like, where it is on the layout and schematic... Yeah I think it's kind of brutal with what we do at [redacted], which is now, there'll be like a rework for [the technicians], and we'll have to like manually label. We have to take a picture of the layout, and then bring it into like some type of editing tool like PowerPoint and then like add arrows to the points that we want to probe... it's like we're passing back like a billion, like little like pictures, and you have to like talk on the phone a lot about it." (P5, Q12)

Two industry engineers shared that communicating with technical staff from the fabrication house was sometimes challenging as often these technicians who actually fabricated, reworked and tested the boards were not familiar with the design itself. In these situations, engineers usually turned to printed materials, annotated images, or emailed presentations to communicate their design intent.

3.2.3 Design Considerations

From this first study, a few primary design considerations were motivated by the feedback shared amongst participants:

DC1 **Reduce context switching and facilitate pattern matching:** Participants expressed the need to move between different representations of the schematic, layout, and board often (Q1—Q4, Q9, Q10). Component, pin, and net localization in particular was cited as tedious since going between information in the schematic and board involved pattern matching with the layout as an intermediary, particularly in dense, complex boards without silkscreen.

DC2 **Show relevant information without cluttering the view:** Participants expressed interest in having context-relevant information accessible through both the design files and also when directly interacting with the board, but were wary of excess visual clutter (Q7, Q8). Some suggested making the display of information optional, such that the user could decide to turn it off.

DC3 **Support habitual and intuitive interactions with the PCB:** While participants each followed slightly different methods of localizing issues and taking measurements, they generally followed a common approach (Q1—Q4, Q7, Q8). An AR system should simplify methods of taking measurements or seeking information, but should not depart greatly from current habits or workflows.

DC4 **Facilitate collaboration:** A few participants noted that finding design elements and following measurement procedures were exacerbated when working with collaborators who were less familiar with a particular design (Q12). A solution that guides the user with relevant information can be extended to help support collaboration (in real-time or asynchronously).

3.3 Interaction Techniques

We derive four core interactions motivated by the needs finding feedback and design considerations discussed above. We then assemble these core interactions as building blocks to support the debugging workflows followed by engineers. We use the term *element* here to refer the circuit elements of a component, pin, or net within the design. We use the term *element identifier* to refer to the component reference designator, pin number, or net name for their respective elements. In this chapter, we will refer to a hypothetical system that implements these proposed interactions as *Augmented Silkscreen*. We will refer to the interview asset we produced and used for evaluation as the *simulator*.

3.3.1 Core Interaction Techniques

The core interactions are categorized by direction of information flow: either from the design files (schematic and layout) toward the PCB, or from the PCB toward the design files (Fig. 3.2).

From Design Files to PCB The two core interactions relating design files to the PCB are *element localization on PCB* and *metadata annotation on PCB*.

Element localization on PCB (ElemLocOnPCB): As per DC1, we found that engineers traditionally follow a two-step process to localize elements on the PCB given a target element on the schematic. First, they textually pattern-match an element identifier to the corresponding one in the layout file using the find command. Then, they spatially pattern-match the layout to the PCB to identify the corresponding PCB element. A few

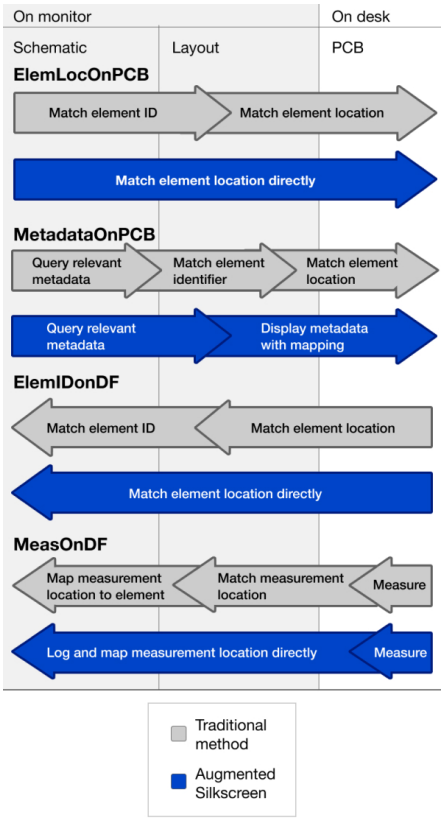


Figure 3.2: *Information Flow of Core Interactions*

ECAD tools [3, 71] support cross-probing between schematic and layout. Using an AR system allows us to extend this interaction to the physical PCB, such that a selection in the layout or schematic view results in an augmented highlight of the matching design element directly on the PCB.

Metadata annotation on PCB (MetadataOnPCB): During the process of debugging, engineers often query the schematic for element attributes that determine the function of the circuit, such as resistor values, IC packaging, diode reverse voltages, or inductor max currents. They keep this knowledge in short-term memory as they subsequently formulate hypotheses for a root cause or look to take their next diagnostic measurement. As per DC1 and DC2, we seek to minimize the cognitive load of keeping information in short-term memory by

bringing this information to the PCB through annotating the PCB element with this element metadata in the view of the user. Additionally, user-inputtable text field annotations can enable users to annotate elements with freeform notes.

From PCB to Design Files The two core interactions relating PCB to the design files are *element identification within design files* and *measurement annotation within design files*.

Element identification within design files (ElemIDOnDF): We learned that engineers follow the same two-step process as described in ElemLocOnPCB in reverse to identify or localize elements on the schematic given a target element on the PCB. Pertinent to DC1, we propose enabling directly making selections on the PCB instead via an interactive probe to select, identify, and localize the same element within the schematic and layout. Probes are commonly used in PCB measurement tasks and are therefore a familiar method of direct PCB interaction.

Measurement annotation within design files (MeasOnDF): Finally, taking diagnostic measurements is a key part of debugging workflows. Augmented Silkscreen would support this interaction by capturing measurement data from benchtop test equipment probed on the PCB and relaying it back to the design files addressing DC1 and DC3. As a practical implementation note, nearly all benchtop test equipment break out their `get` and `set` functions over SCPI/VISA, a standardized measurement instrument API.

3.3.2 Interaction Technique-Supported Workflows

We synthesize these core interaction technique building blocks to support entire debugging workflows.

Diagnostic Measurement Participants described the process of capturing and logging measurements as an important method to assist in deductive root cause analysis. Combining the ElemIDOnDF and MeasOnDF interactions enables users to take a measurement with probes (MeasOnDF), capture the location the measurement was taken (ElemIDOnDF pin),

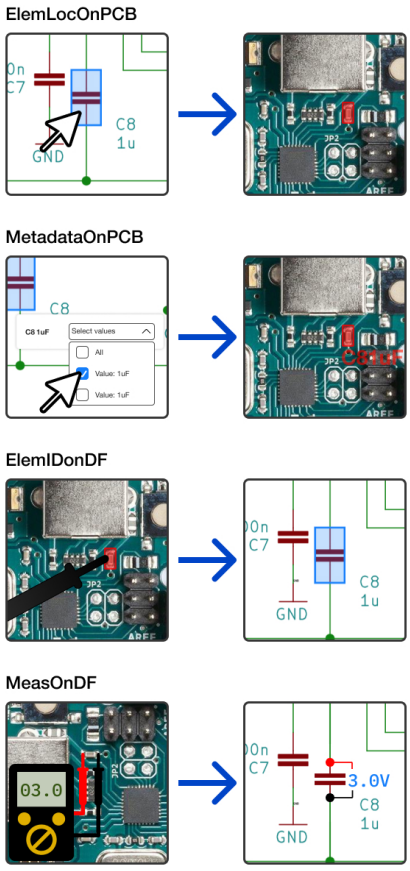


Figure 3.3: *Depictions of Augmented Silkscreen core interactions*

identify the involved nets (ElemIDOnDF net), and include the information on the design file view along with the captured measurement. For example, consider a user that wishes to record measurements, and so starts a new debugging session in Augmented Silkscreen's design file view. The user may take a measurement of a voltage rail with a digital multimeter. From the position of the probe locations on two pins, the corresponding nets for the positive and negative probe terminals are determined via ElemIDOnDF. The measurement value is captured along with its location in the design file view via MeasOnDF.

Bring-up When engineers first apply power to their boards, they typically follow an exacting protocol to ensure all components were assembled properly. By automating ElemLocOnPCB interaction, all uninstalled component locations and all pin one locations (indicative of correct component orientation) can be highlighted directly on the PCB permitting rapid visual checks. Similarly, by entering a set of desired nets to test into the design file view, and optionally providing functional limits, Augmented Silkscreen can sequentially display probe points on the PCB, again via the ElemLocOnPCB interaction. A user may then follow the *diagnostic measurement* technique described above to sequentially capture the measurements back to the design files for comparison to set limits.

Visual Inspection Participants described the need to sometimes query an element's metadata directly within the board view, for example, after noticing a given component's rise in temperature or in determining to which net a certain pin was connected. By combining ElemIDOnDF and MetadataOnPCB the user can select an element directly via probe on PCB and have the metadata annotated directly in the PCB view without having to refer to the design files.

Remote Collaboration To facilitate remote collaboration, many participants pointed out the need to call out to specific elements on the board with a set of instructions. In support of DC4, this can be achieved by splitting Augmented Silkscreen's design file view

and augmented PCB view across two locations. Synchronous collaboration can be enabled if one user (for example, the board designer) has the design file view and the other user (the remote debugger) has the PCB. The designer may select elements such as component or pins (probe locations) to display on the remote debugger's view of the PCB via ElemLocOnPCB. The *diagnostic measurement* interaction may then be used to capture the remote debugger's measurement values back to the designer's design file view. In an asynchronous collaboration, the designer could leverage the MetadataOnPCB interaction to tag elements in the PCB view with freeform instruction call outs. This could be helpful for communicating rework instructions or step-by-step debug procedures.

3.4 Study 2: User Evaluation

To solicit feedback on the interactions we designed, we remotely conducted another round of structured interviews using an interactive simulation. Each interview lasted approximately one and a quarter hours.

3.4.1 Participants and Procedure

For continuity, study 1 participants were re-invited to participate. Six out of the original eight participants were able to join (see Table 1). No additional participants were recruited. The study was divided into three main sections:

1. Feedback on core interactions and interaction-supported workflows
2. Feedback on variations on the attributes of core interactions
3. Timed element localization tasks

Interview Assets To help participants envisage and solicit feedback on our interaction concepts during the remote interviews, we produced two artifacts: a web-based, interactive PCB simulator and a set of envisioning video sketches [161]. The simulator mirrored the

workbench setup described by participants in Sec. 3.2.2. It comprised of two components: (1) an in-browser, interactive schematic and layout viewer on the participant's monitor (Fig. 3.4(A)) simulating the schematic and layout viewer an engineer would have open on their computer during debugging, and (2) an in-browser, interactive top-view PCB image on the participant's own touchscreen device (Fig. 3.4(C)) simulating the PCB the engineer would have on their lab bench during debugging. In order to deliver Augmented Silkscreeninteractions that span design files and board augmentation, the two were linked via a web socket enabling real-time interaction in the schematic and layout viewer to affect augmentations on the mobile PCB view, and vice versa. Participants could select a component, pin, or net in either the schematic, layout, or mobile PCB view (via clicking or tapping via probe) and have the corresponding elements highlighted in the other two views. Additionally users could right click on a component revealing metadata and a button to show that metadata augmented on the PCB view (Fig. 3.4(A). Presenting interactions via this simulation prototype allowed for the users to use their own devices at home and allowed us to easily modify specific attributes of the way the core interactions were presented to users (see Sec. 31). This web socket could also be disabled to test situations without Augmented Silkscreencross-linking between design files and board (see Sec. 3.4.1). For the evaluation, we used the schematic, layout, and PCB image of an Arduino Uno R3[1].

Additionally, to further assist users in visualizing the interactions, we recorded a set of POV video walkthroughs for each interaction technique and each interaction-supported workflow. Each sketch illustrated the schematic and layout interactions in screen capture and view of a desk top PCB in a time-synced inset (ex. Fig. 3.4(D)). The PCB augmentations in the shown videos were projected via overhead projector (AAXA P7[2]). A narration also helped to describe the interaction. During the interview, if the participant was unclear on the video content, the interviewer provided additional description until it was clear.

---

[1]https://store.arduino.cc/usa/arduino-uno-rev3

[2]https://www.aaxatech.com/products/p7-pico-projector.html

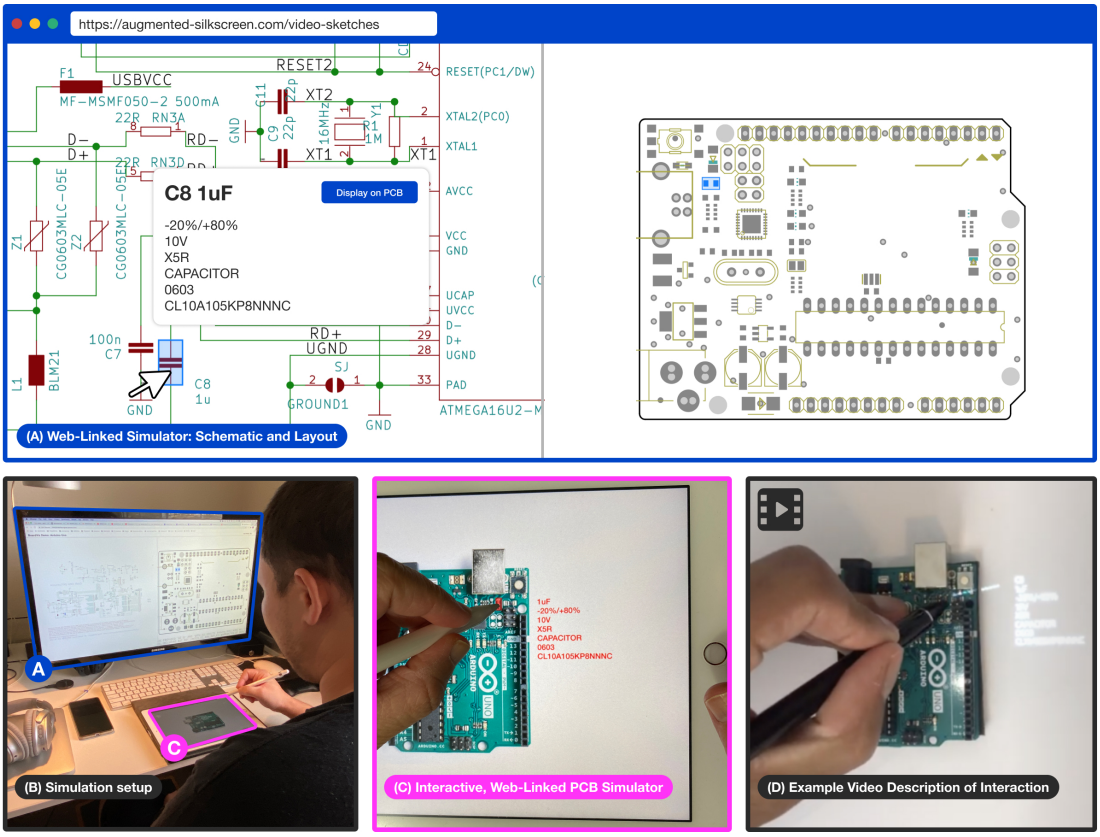


Figure 3.4: *The web-linked simulator consists of two components: an on-monitor design files viewer and an on-device PCB view. (A) In-browser canvases contain interactive views of the schematic (left) and layout (right) of the design, just as engineers would have on their monitor during PCB debugging. Here, the user has selected capacitor C8 in the schematic (A, left), which would cause the corresponding element to highlight in both the layout (A, right) and PCB view (C). (B) A remote participant has the simulator open on their monitor and touch screen device. (C) The PCB simulator imitates a PCB a user would have on their desk. Here, a component is augmented with a box highlight and a metadata annotation. A user can use a probe to interact with the PCB simulation, which would affect the state of schematic and layout (A). (D) Freeze frame of inset from an example video description (Visual Inspection video). The video actually shown to the user contains a time-synced screen recording of design file view (A) with (D) inset picture-in-picture.*

Part 1 Procedure The participant was shown each video sketch and the interactive simulator, starting with core interaction techniques and ending with interaction-technique supported workflows. Between each video sketch, the participant was then verbally asked the following questions:

1. *"Would you find this interaction to be helpful, not helpful, or have no impact on your debugging workflow?"*

2. *"How might this interaction affect your workflow?"*

3. *"How likely would you be to adopt this interaction on a scale from 1 (would not use) to 7 (very likely to adopt)?"*

We analyzed responses via thematic analysis [20], by transcribing the interviews, coding recurring themes, and noting outliers.

Part 2 Procedure To better understand how certain design decisions align with the stated design guidelines, we asked participants to assess variations on attributes of the core interactions in five categories (Fig. 3.5):

1. *PCB Target Visualization:* Per DC2, how does the visual design of augmentation influence element localization in ElemLocOnPCB? Options: (a) Box—filled in rectangle, (b) Circle—unfilled circle, (c) Crosshair—perpendicular, intersecting lines, (d) Layout inset—highlight is shown briefly, an inset segment of the layout local to the target element is projected next to the board

2. *PCB Annotation Method:* Per DC2, what is the preferred length and where is the preferred area for on-board annotations in MetadataOnPCB? Options: (a) None—no annotation, (b) On-board minimal—annotation depicting only element identifier and value projected adjacent to the element (potentially overlapping the PCB), (c) Off-board minimal—only element identifier and value; projected on tabletop adjacent to the

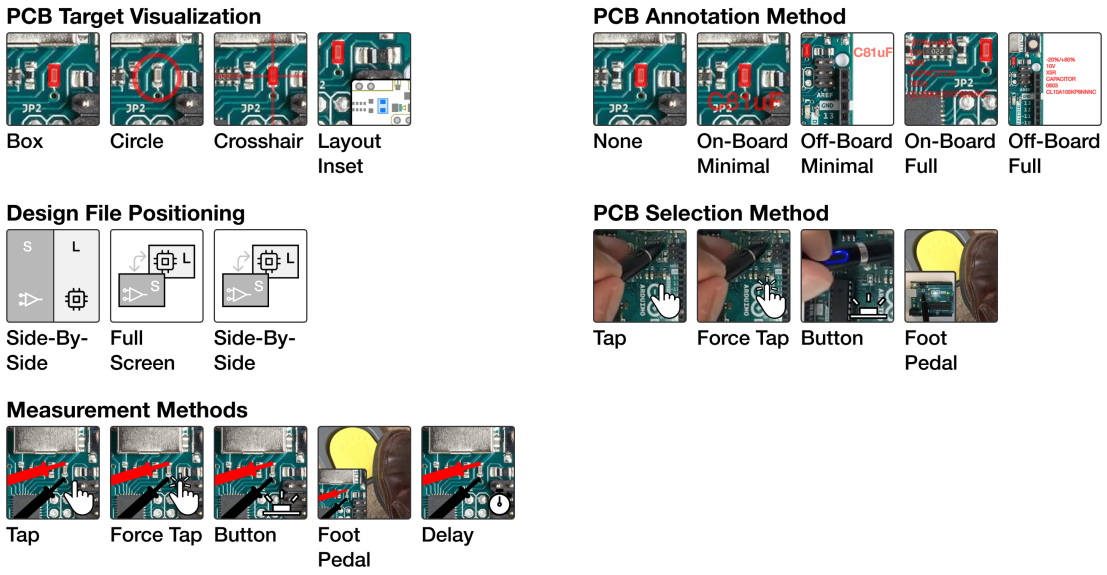


Figure 3.5: *The matrix of variations we presented to participants to elicit design feedback on attributes of the core interactions.*

board, (d) On-board full—all element metadata fields projected adjacent to element, (e) Off-board full—all element metadata fields projected on tabletop

3. *Design File Positioning:* Per DC1, how the positioning of design files on monitor influence element identification within design files in ElemIDOnDF? Options: (a) Side-by-side, (b) Full screen—schematic and layout each took entire screen, flipped between files, (c) Layout inset—layout local to the target element is inset on schematic view

4. *PCB Selection Method:* Per DC3, what is the preferred method to trigger selection of PCB elements with a probe for ElemIDOnDF? Options: (a) Tap—tap top of element briefly to select, (b) Force tap—similar to BoardLab, applying force to probe tip triggers selection, (c) Button—button on barrel of probe triggers selection, (d) Foot pedal.

5. *PCB Measurement Capture Method:* Per DC3, what is the preferred method to trigger a measurement capture on PCB with a probe for MeasOnDF? Options: (a) Tap—

tap pins to capture selections, (b) Force tap—applying force to probe tips triggers capture, (c) Button—button on barrel of probes triggers capture, (d) Foot pedal, (e) Delay—stationary probes for 3 seconds triggers capture.

Items (1), (2), and (3) were delivered via the interactive web simulator. Items (4) and (5) were described with the video sketches. After the demonstration, we asked participants about their general impressions, how they would rank the presented variations, during which workflow they might use it, and why.

Part 3 Procedure Users performed two timed component selection tasks: finding components on the board given a target in the design files, and finding a component in the schematic given a target on the board. An interactive web simulator delivered the schematic and layout on their monitor, and a PCB image stand-in on their touchscreen device in an imitation workbench set up (Fig. 3.4(B)). A standard capacitive stylus shipped to the participants was used to select components on the PCB. For the *find on PCB task*, a target component was highlighted on the schematic and layout. The user's task was to select the corresponding component on the board. When Augmented Silkscreen (cross-linking between design files and board) was enabled, the target component on the board had an augmented highlight as well. For the *find on schematic* task, a target component was highlighted on the board. The user's task was to select the corresponding component on the schematic. Selection cross-linking between the layout and schematic as in KiCAD [71] was enabled as baseline. When Augmented Silkscreen was enabled, the target component on the schematic and layout were highlighted. For each task, all six users performed twenty different component selections: ten selections with Augmented Silkscreen and ten without, with order randomized (for a total of 60 samples per condition minus omissions). Users were permitted a short practice round to familiarize themselves with the selection task. Audio feedback indicated if a user selection was correct or not. Schematic and layout visualizations were kept the same between conditions. Timing started when the component to be found was presented to user (on the schematic and layout in the *find on PCB task* and on the board in the *find on schematic*

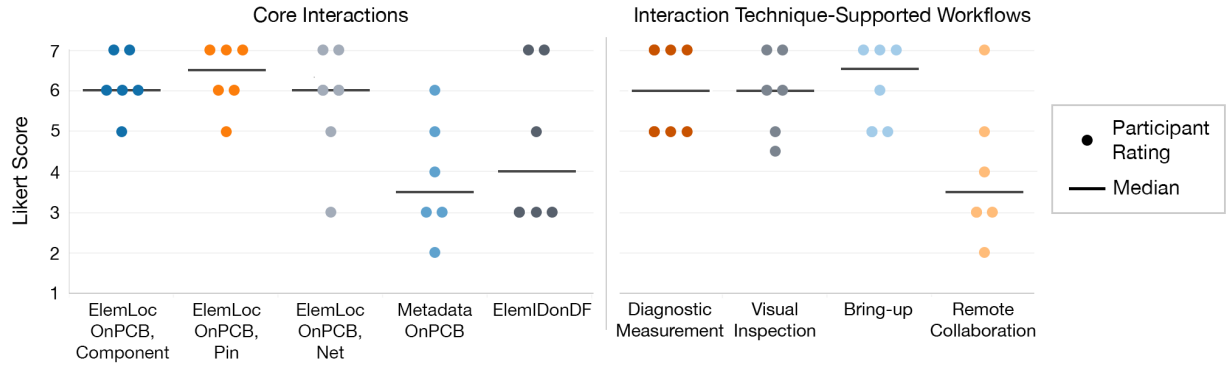


Figure 3.6: *Results from Part 1 (core interaction and workflow feedback).*

task). Timing stopped when the user selected the correct corresponding component (on the board in the *find on PCB task* and on the schematic in the *find on schematic task*). If a user selected the incorrect component, the data was logged as a mistarget; if a user indicated it was due to a fault of the capacitive stylus or touchscreen, rather than a true mistarget, the datum was omitted.

Findings from Part 1: Feedback on Core Interactions and Interaction Supported Workflows Users provided illuminating feedback on their preferences and uses of the core interactions and workflows (Fig. 3.6).

Users unanimously rated EL-PCB favorably for localizing components, pins, and nets. *"It basically saves me an extra step... this allows me to go from schematic to board immediately"* (P1, Q13). *"I have to do it pretty much manually in different pieces of software. So, this would have saved me a lot of time"* (P6, Q14). Specific situations in which EL-PCB would be particularly helpful included working with complicated and dense boards (P1), unfamiliar boards (P2), or boards without silkscreen (P5). P7 also pointed out that *"trying to find specific patterns, especially with things are rotated and whatnot is very error prone"* (Q15) for humans, and that computationally-driven augmentation can help to disambiguate.

P1 cited that there might be practical issues with the precision of an AR system highlighting

very small details (the 500 µm pitch pins of a QFN24 package), and P5 raised concerns with the potential for trust issues if the projections were inaccurate: *"this would be useful but misleading, because yeah what if they got stuck with the wrong part."* (Q16) As P7 pointed out, however, *"as the pitch gets more and more fine, being able to identify the pin doesn't help quite so much, because there's not a lot you can do if it's so small that you can't probe it [anyway]."* (Q17)

Nonetheless, highlighting specific pins connected to a given net was also liked, specifically to identify suitable measurement probe points for a given net: *"[when] looking for a component that was easy to probe without shorting anything else nearby... I don't even have to worry about like looking through the package size on the schematic sheet and trying to find something that's like on the net that's going over different schematic pages. [With augmentation] I can just literally look at the board and say, 'Oh, I got this big capacitor right here that has a good probing point."'* (P6, Q18). Another user looked to use this feature to locate accessible nodes since many are obstructed by features such as shield cans or mechanical cases "in industry boards." (P2, Q19).

As an interaction in and of itself, MetadataOnPCB rated poorly, with many users citing that selecting metadata on the schematic to display within the board view was redundant: *"If I'm already on the schematic and the information is there on the schematic it's not super helpful to display it again on the side of the board."* (P2, Q20). However, when combined with EI-DF to form the Visual Inspection interaction (allowing users to trigger augmented metadata annotations directly from the board as opposed to the schematic) the interaction was universally preferable. *"I'm doing this 24/7, like which pin is which... if I could do something to where I could just be looking at the board, and not have to look away, and just having it projecting on top of it. That would be – oh my god – I would use that like 100% of the time."* (P5, Q21). P2 added that *"in factory environments... this feature could limit the number of different screens or devices I need to carry around."* (Q22)

Diagnostic Measurement workflow was found to be helpful in not only alleviating an individual's own mental demands but also in capturing unambiguous logs for documentation

and interacting with others. *"This would be extremely helpful, because... what I've been doing up until this point is, I may like probe something and then I'll just keep in the back of my mind like okay it's this value. And then I'll just keep like three to four values in my head"* (P5, Q23). *"I've had personally a whole lot of negative experiences of people measuring the wrong things and telling us that they got such and such results."* (P8, Q24).

All participants appreciated the interaction technique-supported Bring-Up workflow: *"very helpful useful during board bring-up, we're just trying to take a basic DC measurement on multiple nodes quickly, smoke test... a lot of times that's tens of nets that we're trying to measure,... and if we can very quickly step through that without having to go back and forth between the board and the PC recording and Excel or whatever, would definitely save a lot of time."* (P2, Q25). *"I could see this like cutting down board bring-up down in time by like hours."* (P5, Q26) Some users suggested they also found it useful to define and project probe points for unstructured debugging: *"[I'm] interested in just the fact that it can highlight the two things I want to probe at the same time though so I know exactly where to put those probes."* (P1, Q27). The explicit probe point projections inspired some users to suggest it can assist train those unfamiliar with their design. *"If I was giving this to like a, like undergrad or like an intern or something you know, probably be pretty useful for them just to quickly catch on"* (P1, Q28). *"If you... can turn this [probe point projection] into an automation program... this would be great at a factory."* (P5, Q29).

Finally, the Remote Collaboration scenario received a varied set of rankings. One user, affected by remote work during the recent COVID-19 pandemic said, *"I think just the ability to communicate very unambiguously; not only like what part, you know, because you could use the reference designators in like an email or a phone call or whatever to tell people what to look at, but, like, in terms of selecting actual points to measure. I think that would be super helpful. Personally, what I've had to do a lot of recently is taking pictures of the manufacturing preview, and then like drawing a circle on what we need and then sending that back and forth on Slack. And so just, you know, in the sense that, that will speed that up a lot."* (P8, Q30). Many users thought the interactions were useful, but were not frequently involved

in collaborative debugging situations. They recognized that they may have growing utility as the pandemic continues to affect workplaces: *"I think [these interactions] can be useful and especially I guess in a situation like we have right now where everyone's work-from-home. And if I don't want to go to a factory, and there's lab techs at the factory, and they're there trying to figure out what's wrong with the board, and they don't necessarily have the expertise about that board, I could probably walk them through it and highlight stuff... I could see it useful in that kind of situation but it's more of a hypothetical because it's not something that I've really done much yet myself."* (P6, Q31). One user also expressed that it'd be helpful to point out to their software colleagues certain buttons, switches, and plugs to interact with on their development kit, but they are not often in situations where the other user would be actively probing pads.

Findings from Part 2: Feedback on Core Interaction Attributes Users ranked target visualizations for ElemLocOnPCB. Crosshair was the primary choice, but box was seen as nearly as good. *"The crosshair makes it super easy to individually pick out which one is which"* (P7, Q32). *"box one is my favorite because it superimposes the most accurately on the part of interest"* (P5, Q33). A user suggested that a crosshair transitioning to a box is good for initial targeting and reducing visual clutter. The circumscribed circle was missing the details of the component's contour, reducing visual precision and as a result being ranked lowest among most participants. *"The circle is misleading because it's like encompassing multiple parts"* (P5, Q34). Towards this end, an AR system must be precise enough to provide unambiguous augmentations on PCB elements (which can be less than millimeter square for the smallest pins). To help resolve ambiguous cases, a local inset of the layout was appreciated as visual confirmation, but users looked for it to be combined with an on-board visualization rather than as a standalone method.

Toward annotating metadata on the board (MetadataOnPCB), participants generally preferred off-board annotations (indicating that on-board annotations felt cluttered), but there was disagreement on how much information to present—between our options presented,

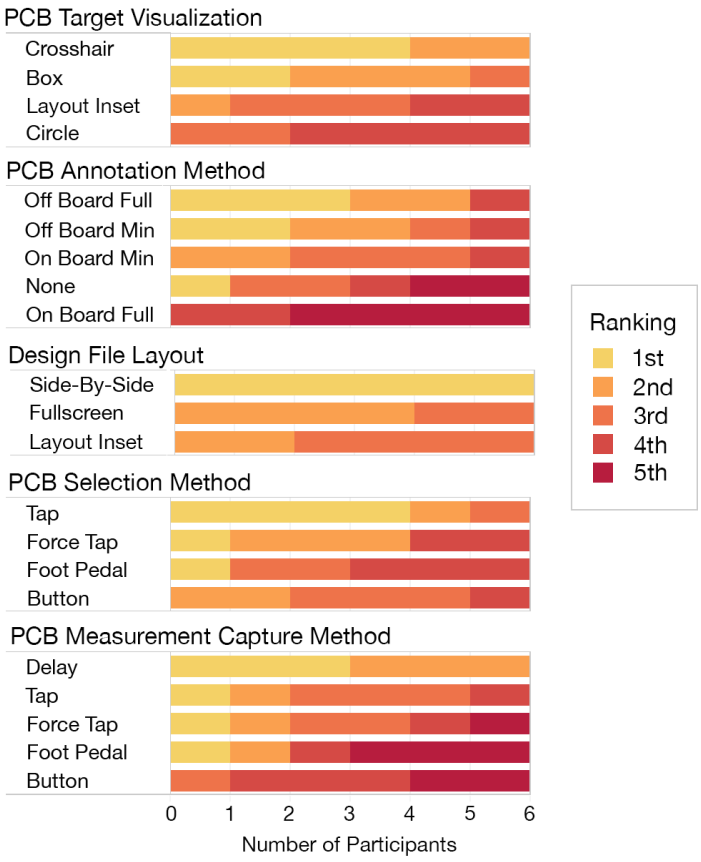


Figure 3.7: *Results from Part 2 where users ranked their preferred variations of core interaction attributes.*

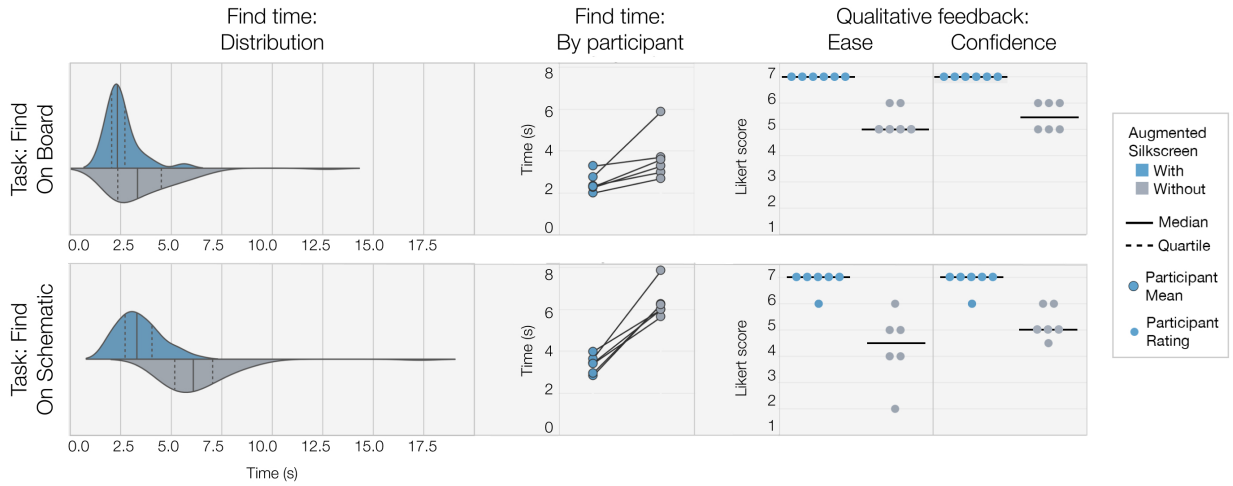


Figure 3.8: *Results from Part 3 where users participated in a timed element localization task.*

there was an even split between all metadata or none, suggesting there is likely some ideal middle ground. Some users indicated their preference would be to control what metadata would be presented. Uniquely, P7 preferred no annotations, but if choosing one, preferred to have a minimum amount of information right on the board, citing that it yields the shortest distance between the component and annotation.

For identifying elements in the design files (ElemIDOnDF), we asked users their preferred design file view to better understand if having an augmented view of the board changed their current design file habits. All preferred to maintain a side-by-side view of the schematic and layout simultaneously, screen real estate permitting, but also looked to have a full screen options as well. Two participants saw value in the layout pop up: *"I do like the spirit of the peek when you click on it, especially if you... want to try and get your bearing with where the component is on the board, but I feel like if you have the crosshair you don't really need that so much"* (P7, Q35). On the other hand, some participants felt like the inset covered information in the schematic. One user indicated that the augmented board view was usable enough, it could eliminate their need to have the layout view on their screen. *"If I was debugging, every*

time I needed to find a component I would use this feature, there's no reason I would look at the board file if I had this feature" (P7, Q36).

For the on-board element selection method in support of ElemIDOnDF, most of the participants indicated, if technically feasible, a simple light tap was preferred. *"The most intuitive one of the best is just tap to select with minimal force"* (P1, Q37). Multiple users worried that a force tap could damage small components, and that pressing a stylus button could cause probes to slip off small probe points. One participant liked the foot pedal selection the most, citing that it allows them to place probes carefully eliminating situations where components can be shorted or damaged.

Amongst the methods to trigger measurement capture (MeasOnDF), delay was the almost universal preference, as it matches the natural use of a multimeter (waiting for the measurement to settle). *"I feel like the way just a normal multimeter works is very intuitive, you just tap on things and like sometimes it takes [after] there's a delay on the screen."* (P1, Q38). One user (P6) ranked foot pedal at the top of the list, as they felt it was deliberate while allowing precise positioning of probes in both hands.

Findings from Part 3: Timed Element Localization Task On average, users performed the *find on board* task 31% faster with Augmented Silkscreen compared to without, with a mean difference of 1135 ms across all samples (per-sample t-test, $t(58)=4.31$, $p<.001$; per-participant Wilcoxon Signed-Rank, $Z=0$, $p=0.031$). True mistargets fell from 16.94% without Augmented Silkscreen to 8.47% with Augmented Silkscreen. Users rated ease, rose from a median of 5.0 to 7.0, and confidence rose from 5.5 to 7.0, on the 7-point scale. *"I was very confident [with Augmented Silkscreen] because generally I knew what I was looking for, and also the highlight was basically telling me it, I didn't need to double check it most of the time... Without the highlight, I'd have to look at it, choose which one I'm pretty sure it was, double check, and then go back and click once I was confirmed what I thought was."* (P6, Q39).

In the *find on schematic* task, users were, on average, 46% faster with Augmented

Silkscreen, with a mean difference of 2923 ms across all samples (per-sample t-test, $t(59)=10.1$, $p<.001$; per-participant Wilcoxon Signed-Rank, $Z=0$, $p=0.031$). True mistargets were 3.33% for both conditions. Users' ease and confidence score increased to a median of 7.0 for both metrics, from 4.5 and 5.0 respectively, out of 7 points. *"If I have to click it on the layout and have it show up on the schematic, yeah, that's helpful compared to what I have... it already, you know, just saves me the step of switching windows essentially, in searching."* (P7, Q40).

We note that the selection task given may have been too easy (as evidenced the by high ease score for the baseline) with a single page schematic and small, low component count board relative to what is typically found in a commercial product. A more complex design (with greater number of schematic pages and higher component count) may have yielded a starker difference between control and condition with Augmented Silkscreen, with the control more likely to take tens of seconds to minutes to localize a given component as per the qualitative feedback during needs finding. *"Definitely would be significantly easier with the AR link just because we're navigating like 55 page schematics as opposed to this simple one pager here with a really simple layout, so it's much more difficult to keep your schematic and board view aligned... It's a lot more like zooming on the board side, page changing on the schematic side, and then cross referencing to a real PCB."* (P2, Q41).

3.5 Discussion

Through a three-part study, we have explored the design space of using AR visualization and interaction as tools for assisting electrical engineers with their PCB debugging workflows and preliminarily evaluated a proposed set of interaction techniques. In particular, we found that four specific tasks benefited the most from our proposed interaction techniques:

1. finding components (ElemLocOnPCB components) and probe points (ElemLocOnPCB pins and nets) on the PCB,

2. immediately providing element metadata at the board without referencing design files (Visual Inspection)

3. logging of unstructured measurement queries with associated probe points (Diagnostic Measurement)

4. unambiguous, spatially co-localized, and potentially automated probe point visualizations for directed measurement workflows (Bring-up)

In support of DC1, participants' most cited reason for their expectation of increased efficiency was confirmed to be the reduction of context switching between files. For example, ElemLocOnPCBremoved out the need to reference layout when moving from schematic to board (Q13, Q14, Q15), Visual Inspection cut out the need to flip from board to schematic to pull metadata information (Q21), and Diagnostic Measurement and Bring-Up workflows took out the need to flip to instrument panels and logging documents (Q25).

Careful design is needed towards maintaining a fine balance between providing relevant information and avoiding a cluttered view (DC2) and warrants further study. While users generally agreed that information should be provided out of the direct line of sight to the PCB, there was disagreement on how much is helpful (see MetadataOnPCB Findings). Breaking out control to users for them to adjust based on their context may be best.

The choices amongst users affirmed that supporting habitual interactions with the PCB (DC3) is an important design consideration, as evidenced by responses to the preferred PCB element selection method (simple tap, Q37) and measurement capture method (delay, in line with current behavior, Q38). Users were enthusiastic (Q23, Q25, Q26) about interaction-supported workflows that directly mirrored and supported their existing practices (e.g. Diagnostic Measurement, Bring-Up).

Finally, users generally were interested in how the techniques could help support collaboration (DC4), but for only one did the use case arise frequently enough to say they would adopt it (Q31). However, these situations may be increasingly common with a progressively more globalized electronics manufacturing pipeline, a stay-at-home pandemic, and decreasing knowledge barriers for participation in electronics design.

Feedback from participants elucidated a few practical challenges towards the construction

of a future system regardless of how augmentations are delivered (head-mounted device, handheld mobile video pass-thru AR, projective AR, or other). First is the need for extremely precise and stable, board-locked visualizations. Users expected the system to be able to augment the smallest pin that can be reasonably probed, or a methodology to disambiguate imprecise visualizations. Second is the need for accuracy in board-locked visualizations. Users expressed that they would mistrust the system if it could not provide accurate overlays (Q16). Furthermore, PCB modifications during debugging such as reworked components or breakout wires may also cause the element on the board to no longer align with the design files. A function to support deviations from the imported design files could address this. Finally, on the wish list for one participant was a system that could be easily portable to a number of environments, such as the factory (Q22).

3.5.1 Limitations and Future Work

Due to the COVID-19 pandemic, studies were conducted remotely prohibiting the ability to collect observational data. Following the pandemic, we would look forward to observing user workflows directly in a simulated debugging task, beyond the video call with video prompts and a PCB simulator we used in this evaluation. We could perform the timed evaluation on a real, potentially more complex, PCB than the web simulator board which is limited to the real estate of the user's touchscreen. One participant did note, however, of our simulator, *"I felt like basically your emulation setup was pretty representative of like how the tool would be in in real life . . . it carried over pretty well."* (P6, Q42). Future work could extend the timed element selection tasks in this chapter toward a more general and open-ended timed debugging procedure allowing for multiple proposed interactions to be leveraged. The small number of participants may limit the generalizability of the findings. While we received interesting feedback from our relatively small sample size, a larger study could yield greater variety and nuance in the discussion, especially on topics where the responses were less uniform (e.g. MetadataOnPCBand Remote Collaboration). It would also provide larger effect sizes in the analysis of the quantitative data. Users cited that referencing datasheets is a common source

of debugging information as well. Work towards parsing and linking component datasheets to be able to provide context-relevant data would further help to decrease context-switching. Only click- and touch-based methods of element selection and file navigation were considered for this study, but some users expressed interest in multi-modal methods combining voice or gaze.

While not explicitly a debugging procedure, users frequently commented that the methods proposed in this work can help speed up and decrease errors in assembly and potentially validation, warranting more thorough investigations toward these use cases. Augmented Silkscreen may also be used to streamline assembly workflows by sequentially highlighting the locations of each component installation location via ElemLocOnPCB, however a Bill-of-Materials (BOM) view is likely needed to help provide an ordering to the installation procedure. Engineering validation is a process in which board revisions are tested against a set of functional requirements. Augmented Silkscreen could be used to assist in preparing samples for test which can be a manual process, but as these tests often must occur on a large sample set of the population, automated test equipment is typically leveraged.

Finally, we are excited for future work to incorporate these interactions and feedback into a deployable augmented reality system. The system would comprise of three parts: (1) a graphical program running on the user's computer that presents the schematic and layout, (2) an augmented reality system that can deliver augmentations on the PCB, and (3) a probe to track the user's interactions with the PCB. To be practical, the program would need to be built as an extension of an existing ECAD tool or as a separate application able to ingest and parse ECAD schematic and layout documents. For delivering board augmentations, multiple methods of delivering AR augmentations are possible: via headset (as in Hahn et al. [54]), via see-through mobile AR (as in InspectAR [61], or via projective augmentation (as in Mascot [131]. The board itself can be tracked via computer vision (as in InspectAR) or fixed in known location (as in Mascot). A probe to interact with the board could be tracked via magnetic tracking (as in BoardLab [49]), computer vision, optical tracking, or mechanical linkage. To the best of our knowledge, a system tying these three components together has

not yet been developed. Doing so would allow for the interactions proposed in this chapter to be realized.

3.6 Conclusion

In this chapter, we proposed Augmented Silkscreen, a set of augmented reality interaction techniques to assist electrical engineers in PCB debugging. We find that combining augmented visualization and augmented interaction on printed circuit boards unlocks promising avenues to alleviate the frequent context switching and spatial pattern matching exercises required by engineers' current ECAD tools. For experts, this can lead to more efficient debugging. In timed element selection tasks, this led to a 31% and 46% decrease in time to find a given component on the PCB and in the schematic respectively, with potential to decrease element localization more drastically in more complex board designs. For those unfamiliar with a PCB design or PCB design in general, the unambiguity of WYSIWYG augmentations on the board directly can help to make basic PCB workflows more accessible. While the bulk of the effort by the HCI community has focused on supporting the latter group, we hope this work will inspire more work toward supporting hardware workflow challenges for both maker and expert populations.

Chapter 4

BUILDING THE AR DEBUGGING WORKBENCH

In this chapter, we apply the design principles extracted in the previous chapter to build a prototype spatial computing system for PCB debugging. We call our system ARDW: Augmented Reality Debugging Workbench. This chapter was published at ACM UIST 2022 [30]. An illustrative video can be found at: `https://youtu.be/RbENbf5WIfc`.

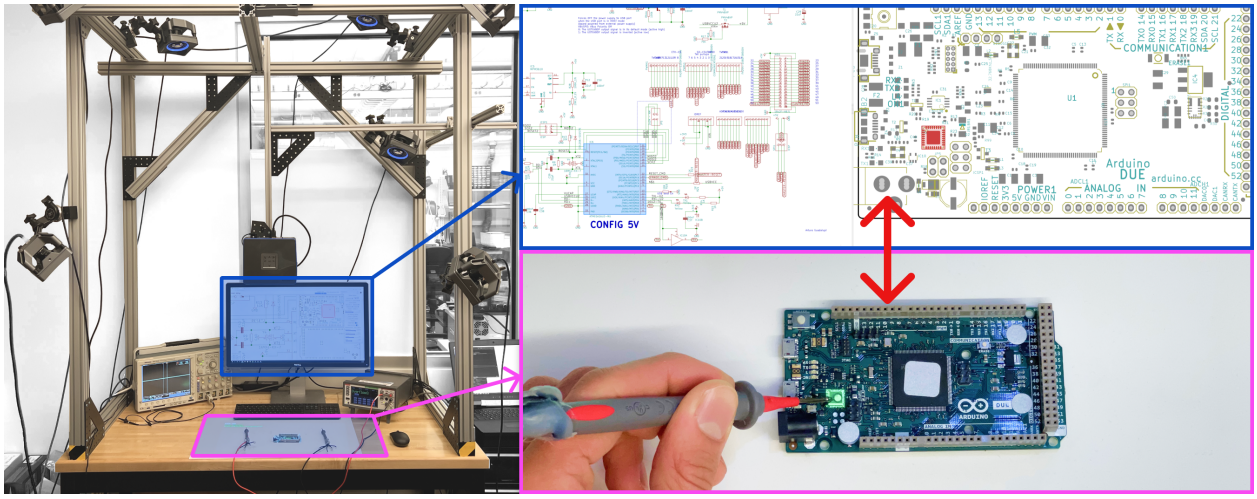


Figure 4.1: *ARDW is an end-to-end system (left) that enables cross-linking between the physical printed circuit board (bottom right) and the design files on a PC (top right) via projected AR (green highlight, bottom right). When a user selects a components, pins or pads on the PCB, the corresponding element gets highlighted on the design files and vice versa (right).*

4.1 Introduction

As mentioned before, the increasing ubiquity of technology has been spurred by rapid advances in electronics manufacturing over the past few decades. With developments such as surface mount technology (SMT) and integrated circuit (IC) packaging, printed circuit boards (PCBs) have become increasingly dense and complex featuring tens, hundreds, or even thousands of components and double-digit layer counts.

Meanwhile, the tools to support engineers in debugging faulty PCBs during the design phase remain mostly unchanged — engineers probe various nodes on the PCB with test equipment while referencing the schematic and layout diagrams to reason about their circuit.

While the human-computer interaction (HCI) community has mainly focused on supporting makers, prototypers, hobbyists, and students with tools to help support the debugging of low-volume breadboard designs [176, 145, 77, 68], there has been less work toward supporting the workflow of circuit design and development moving towards production [70]. We seek to add to the small but growing area of HCI literature focusing on PCB design [146, 49, 93]. Finally, an increasingly globalized supply chain, design outsourcing, and pandemic-normalized remote work have each introduced new challenges in debugging PCB designs collaboratively.

In the previous chapter, we discussed that electrical engineers are interested in augmented reality (AR) as a paradigm to help reduce the cognitive load, accelerate common tasks within debugging workflows, and enable remote collaboration [28].

In this work, we build and present ARDW (Augmented Reality Debugging Workbench), an open-source, end-to-end system that enables cross-linking between virtual design files and the physical PCB via projected AR and tracked test probes. At its core, ARDW enables *augmented visualization* by facilitating for selections in the design files to be highlighted on the physical PCB, and *augmented interaction* by allowing for selections and measurements on the physical PCBs to be carried to the design files. In this way, ARDW is the first system to provide an augmented *bidirectional* cross-link across schematic, layout, and physical PCB. We conduct a study with ten electrical engineers using the tool across a set of board

navigation, bring up, and simulated debugging tasks. All participants verified that the reduced context-switching between the PCB and design files allowing for users to more efficiently localize items on the PCB and capture measurements, and making their debugging experience more seamless. Finally, building on the user feedback from the user study, we offer design considerations towards producing future AR systems for PCB debugging.

4.2 Workflow and System Features

We introduce the features of ARDW through an an exemplary PCB development scenario. The walk-through illustrates a few common issues electrical engineers may encounter during a typical development scenario and discusses how ARDW can help to assist the electrical engineer in debugging them. Figure 4.2 and the video demo (https://youtu.be/RbENbf5WIfc) provide images and clips to help visualize these interactions. All features discussed in this section are implemented in our system. See section 4.3 for full implementation details.

Kofi is designing an 8-layer PCB for a microcontroller development kit in KiCAD. The design is in the early Engineering Validation Test (EVT) phase of the hardware development process, where quantities are low, between 50 and 500 units.[1] Kofi and his team have received the shipment of 70 assembled boards from the fabrication house. Kofi has a ARDW system installed in his lab's workspace. ARDW generally mirrors the set up of other workbenches in the lab with a rubberized, anti-static mat on the table surface, a PC, monitor, keyboard, and mouse, and a benchtop digital multimeter; however, it also has a few additional components: a set of eight overhead-mounted tracking cameras and overhead mounted projector (see fig. 4.4).

4.2.1 Loading Design

To start, Kofi first exports his design from KiCAD into ARDW's viewer using ARDW's import utility, which integrates into KiCAD as a third-party plug-in (see Fig. 4.2A). The plug-in

[1]For more information on hardware development and release process see: `https://instrumental.com/resources/factory/hardware-engineers-speak-in-code-evt-dvt-pvt-decoded/`

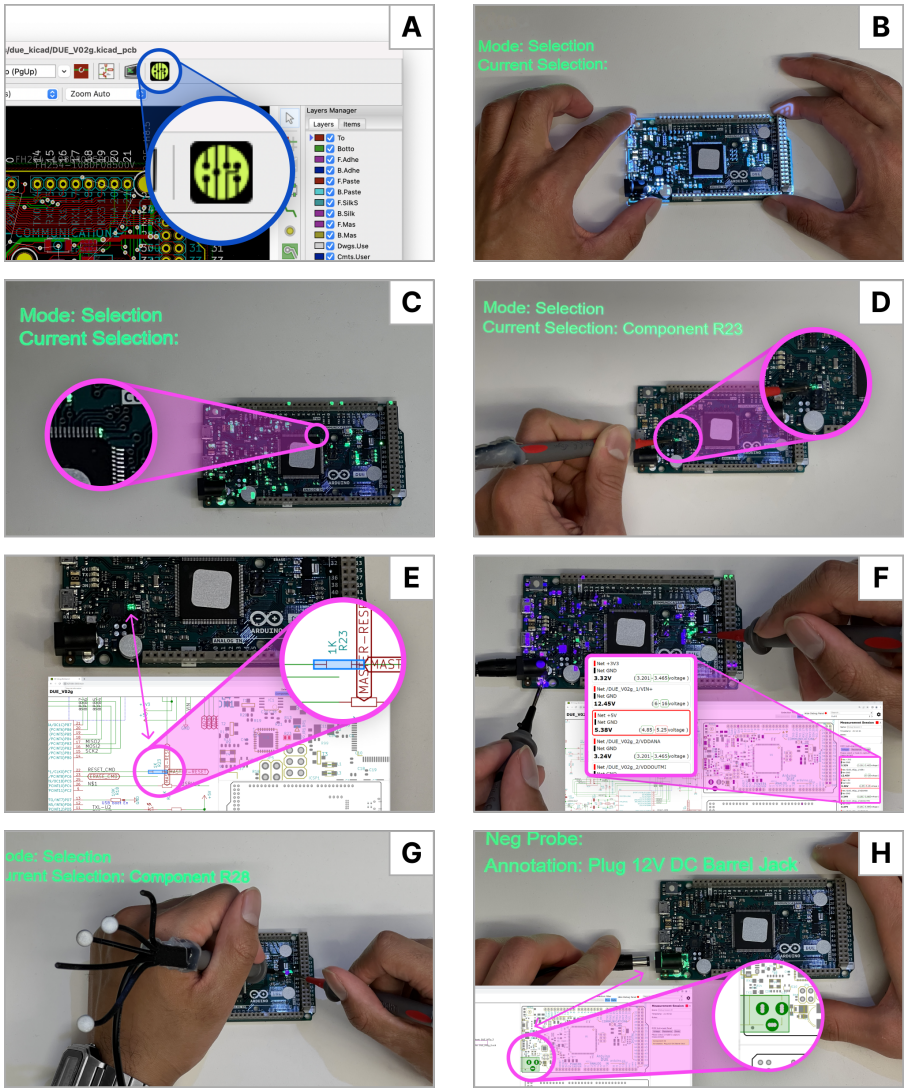


Figure 4.2: *(A) users can bring in their file from KiCAD using ARDW's import utility, (B) user manually aligns PCB to projection, (C) user leverages ARDW to highlight all pin 1s, including the pin on this IC, (D) user employs tracked probe to select components or pins by dwelling on the components, (E) by selecting in any view, such as the board, the item in the schematic and layout are both highlight, (F) when a user specifies a guided measurement in the measurement panel, the probe points are highlighted, the user can take the measurement and it is automatically captured in the measurement panel, (G) selecting an item in the schematic highlights the component on the board allowing users to find the correct item quickly, (H) users can provide guided annotations which are particularly helpful when collaborating with others remotely.*

parses `.svg`s plotted from KiCAD's schematic editor as well as the netlist files produced during the PCB's design phase. This loads the schematic sheets and the board's front and back views into the on-screen interface of the workbench's monitor. By default, the schematic and layout sit side-by-side, however if Kofi's screen real estate is at a premium, it is possible to resize views or just see a single view.

By default, an outline of the board's edge cuts as well as the pads for all components are projected on the anti-static mat on the table surface via the overhead mounted projector. Kofi aligns the board with the projected outline, or, if preferred, attaches stick-on tracking markers to flat areas of the PCB to permit the board to be tracked by the overhead cameras (see Fig. 4.2B).

4.2.2 Visual Inspection

Kofi first starts with a manufacturing check on a subset of the delivered boards. Kofi first seeks to check that the "Do Not Populate" (DNP) components are correct.[2] He selects the "DNP" filter which highlights all components on the board that should not be installed. Typically, Kofi would need to look between a layout diagram where DNP components are highlighted and board multiple times to ensure that each component is not installed appropriately. On smaller or more mature boards this number can be a handful of components, but on larger or early development designs this may be tens of components. With ARDW, Kofi is able to immediately check the appropriate components are not installed.

Next, Kofi looks to confirm the orientation of all functionally-asymmetric components such as electrolytic capacitors, diodes, and ICs, are correct. They select the "Pin 1" filter in ARDW which highlights the leading pin of each component (see Fig. 4.2C). Kofi ensures the package's pin 1 marking matches the highlighted pin 1.

Finally, Kofi performs a visual inspection of the board looking for board damage,

[2]also known as "nostuff", "NS", "unstuffed", engineers will instruct the assembly house to omit populating components to disconnect unused sub circuits or provide optionality to modify functionality without a board re-spin

anomalous component placement, cold or cracked solder joints, or other manufacturing issues. He finds a tombstoned resistor.[3] Kofi selects the component by placing the tip of his tracked probe on top of it and holding for half a second (see Fig. 4.2D). The component is highlighted via projected augmentation as well as in the schematic and layout views indicating selection (see Fig. 4.2E). Kofi can immediately see in the schematic view that the resistor is in series with the enable pin of a critical IC. He can observe in the layout view that the highlighted footprint differs from other adjacent resistors of the same package which likely caused the issue. He re-solders the component by hand, and notes that this incorrect footprint will need to be rectified in the next revision. He cycles through other boards while keeping the resistor highlight on to spot-check that area on other boards, finding the same issue occurring on two other boards.

Without ARDW, Kofi would typically see a fault on the board, such as a tombstoned resistor, and then visually pattern match that area of the board with the layout diagram to determine the damaged component's reference designator (for example, `R238`). He can then use this reference designator or cross-probing to find the corresponding component in the schematic, and determine the component's function. Often small components and dense boards do not include silkscreen with the component's reference designator. With ARDW, moving between the representations is accelerated.

4.2.3 Bring up

Next Kofi performs a bring-up procedure for his boards. During bring-up, engineers typically step through all the major power rails on their boards, first probing for resistance (to ensure no shorts to ground). They then apply power and measure the voltage of each rail to ensure each rail are within an expected voltage bound, often recording their measurements in a prepared spreadsheet. With ARDW's measurement sidebar (Fig. 4.2F, right), Kofi pre-loads a set of rails he is interested in measuring, the type of measurement, and optional test bounds.

[3]when a two-pin package only solders on one side causing the other terminal to float above its pad

He then clicks "record." ARDW starts with his first specified measurement—a voltage measurement between net `VSYS_3V3` and `GND`. The system highlights all the pin locations for the `VSYS_3V3` net in green (the color of the dot tracked under the positive probe) and the pin locations for `GND` in purple (the color of the dot tracked under the negative probe) (see Fig. 4.2F). Kofi can immediately identify the most convenient location to take the measurement. ARDW is connected to his digital multimeter via the VISA API allowing the system to automatically set the instrument into voltage mode. Kofi places his probes on the highlighted pads and holds for half a second. The system recognizes the location of the probes to match the currently-specified measurement, captures the voltage, and records it to the debug panel, highlighting if the measurement panel is outside the specified test bounds. The system moves to the next guided measurement panel and displays the next set of measurement pin or net locations. We call this mode "guided measurement." In this manner, Kofi is able to efficiently move through the set of required measurement test cases.

4.2.4 Free-form debugging

After bring-up, Kofi finds that a net, `+5V`, is sitting slightly higher than its upper test bound on three of the boards he has tested. He selects one of the faulty boards, and traces the power path to localize the issue. He starts at the barrel jack input, clicking on the jack's positive pin in the schematic. This highlights the corresponding pin on the board. He chooses to record his measurements, selecting the record button in the measurement panel. He uses the system's probe selection filter, deselecting components and nets, to filter only by pins. This way, when he takes the voltage measurement, the system records the pin name locations and associated measurement. He finds the voltage to match his expectations. He continues down the power path repeating the set of above steps, using the schematic to guide his series of measurements. Finally, he arrives at the output of the 12V-to-5V buck converter IC, noting the higher-than-expected voltage on the output despite the expected voltage on the input. After localizing the issue to the buck converter sub-circuit, he accesses the IC's datasheet and determines the buck's feedback network uses a resistor divider to set the output voltage.

He removes power from the board, and puts the DMM into resistance mode. Selecting the resistor in the schematic highlights the component on the board (see Fig. 4.2G). He probes across the resistor and notes the top resistor in the resistor divider is 5% lower than its nominal value listed in the schematic. He takes the same measurement across a random sample of boards finding that the faulty boards all have a similar anomaly. He realizes that he mistakenly approved a manufacturer-requested resistor substitution for a ±5% tolerance component, whereas originally his design called for a ±1% tolerance part. As opposed to guided measurement described in the previous section, free-form debugging allows for the engineer to perform root causing on-the-fly and for their measurements to be automatically logged.

4.2.5 Collaboration

To validate his hypothesis, Kofilooks to perform a rework on ten boards and validate that the rail sits within specified test bounds. However, Kofi's rework team sits in a different office. He sends ten boards to the rework team's office which has an ARDW system in the rework lab. Traditionally, Kofi would take a screenshot of his layout and email an annotated image of the rework request to the rework technician, Deepali. With ARDW, Kofi instead is able to provide a set of annotated instructions to be projected directly on the PCB (see Fig. 4.2H). Additionally, due to ARDW's client-server architecture, Kofi and Deepali can share a session, facilitating bidirectional remote collaboration in real-time. While on a call, Kofi and Deepali launch a shared session, Kofi highlights the mis-toleranced resistor by selecting the component on his layout, Deepali sees the corresponding component highlighted on her board. Observing the board directly, Deepali sees a tall plastic header next in the vicinity of the resistor, which impedes access with a hot air gun to replace the close-by resistor. Noting the header is non-critical, Kofi permits the rework engineer to remove the header as necessary. This type of reasoning about the physical board can be performed quickly with a shared ARDWsession, whereas such a realization and discussion might take multiple exchanges if sharing only 2D screenshots of the board as it typically done.

Kofi prepares a set of annotated instructions for Deepali to validate the fix, following the rework. Some instructions, like plugging in power to the DC barrel jack or toggling a specific switch on the board are associated with a component, while some instructions like specifying a guided measurement between the buck IC's output pin and ground are associated with pin locations. As Deepali performs the rework, ARDW projects the steps sequentially on the board reducing ambiguity and the chance for error. When they perform the specified measurement, that measurement is automatically captured and can be exported to a log to share with Kofi. This function helps the design electrical engineer to delegate repetitive validation procedures to others.

4.3 Implementation

Our system has three main components. The *import utility* is used to load designs from KiCAD, an open-source ECAD tool. The *screen interface* on the workbench's monitor delivers the schematic and layout views as well as the bulk of the visualization control of the system. Finally, the *augmented interface* in turn consists of three sub-parts: a *tracking system* for tracking the positions of test instrument probes and the board, a *projector display* for providing augmentations on the physical PCB, and the *connected test instrument* for capturing measurements. These three main components are linked via a Flask [4] server backend. The flow of information within the system can be seen in Figure 4.3. All of our code is open-source and can be accessed here: https://github.com/ubicomplab/ar-debug-workbench.

4.3.1 Import Utility

The import utility is a plugin for KiCad 5.9, an open-source and free ECAD tool. The plugin is written in python 2.7 and runs in KiCad's layout editor. For schematic data, the plugin consumes SVGs of each schematic page from EESchema, KiCAD's schematic capture application as well as component library (`.lib` and `.cache-lib`) files for schematic symbol

[4] `https://flask.palletsprojects.com/en/2.1.x/`

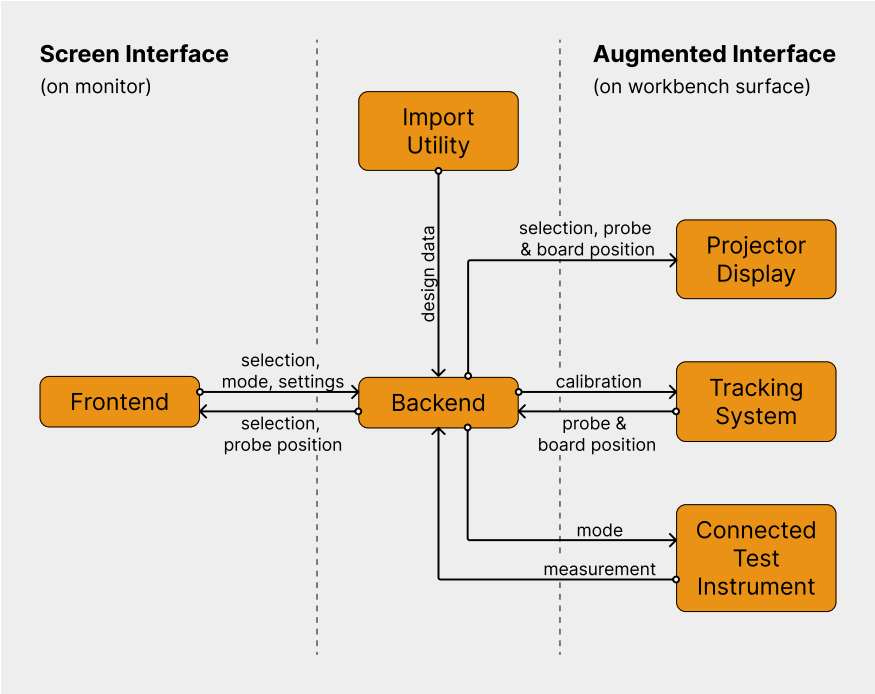


Figure 4.3: *Flow of information in the system. ARDW consists of three main parts. The import utility imports all the design files, the screen interface renders in-browser and displayed the design files and settings to the user, and the augmented interface combines the tracking system, the projector display and the connected test instruments to let the user interact with the board. All these components are connected to the server backend.*

hitboxes. Next, our utility cleans the SVGs and collects component metadata data from the schematic file (`.sch`). We collect net metadata from the netlist (`.net`) file. For layout data, we extend IBOM's KiCAD plug-in [124] which uses an API within KiCAD's layout editor to collect and organize layout data into a format that can be rendered via HTML5 canvas. All above data is finally passed to the server backend as json files, `schdata.json` and `pcbdata.json`, which further processes them, including matching all components, pins, and nets across schematic and layout data structures via reference designator or net name.

4.3.2 Screen Interface

The screen interface is a web application driven by the Python 3.7 Flask-server backend. It is rendered in-browser and displayed on a 27-inch monitor in front of the user.

The frontend is written in Javascript and is connected to the backend via Socket.IO[5], which is used to communicate events such as selections and settings changes, but also enables rendering the probe and board positions in real time. This also means that several windows of the interface can be open at once. For example, if a user has multiple monitors, they can have the schematic open in full screen on one monitor and the layout in full screen on the other. We facilitate remote collaboration by using ngrok[6], a tool that can temporarily exposes the localhost to the internet, connecting multiple computers with full selection cross-linking and augmentation synchronization. When collaborating, both screen and projection interfaces are mirrored. When any user probes or selects, the action is reflected across all users in the same session. As in an individual session, guided measurements and guided annotations can be authored via the measurement sidebar.

The interface consists of (1) a schematic and layout visualizer, (2) a search bar, and (3) a measurement sidebar (see Figure 4.5). We extend IBOM Visualizer [124], which renders and supports interactions with the PCB layout, including metadata such as pin 1 and DNP designations, by also displaying the schematic sheets. As preferred by electrical engineers [28], the schematic and layout sit side-by-side and take up the majority of the screen, but can be customized to the user's liking. The layout is rendered in an HTML5 canvas using design file data in `pcbdata.json` from the import utility. For performance reasons, visuals that are frequently updated, such as selection highlights and probe locations, are drawn in a separate canvas layer above the layout. The schematic is rendered primarily through the SVGs from the import process. Only the selection highlights are rendered in an HTML5 canvas above the SVG, using design file data in `schdata.json`.

---

[5] `https://socket.io/`

[6] `https://ngrok.com/`

Board elements can be selected in both the schematic and the layout by either clicking on them directly, or selecting them in the search bar. In many cases, clicking on an element in the schematic or layout will hit several hitboxes at once, which is resolved by a popup disambiguation menu. Our system supports cross-linking between board representations, meaning any selection made in one representation will also appear on the others, including with augmentations on the physical board.

The measurement sidebar can be opened to use connected test instruments with ARDW and contains a handful of distinct functionalities. The record button at the top allows the user to change the augmented interface between selection mode and measurement mode. The DMM instrument panel imitates the display of a digital DMM, allowing users to change the mode of the DMM to voltage, resistance, or diode and see the current value being measured. Users can also create and run measurement sessions as part of a bring-up or free form debugging workflow.

4.3.3 Augmented Interface

The augmented interface is supported by three main subsystems: (1) the tracking system calculates the pose of both probes and the board in real time, (2) the projector display provides augmentations on the physical PCB and the surrounding workbench surface, (3) and the connected test instrument captures measurements. In the following subsections, we elaborate more on how these subsystems work.

Tracking System We use an 8-camera optical motion capture[7] system (calibrated accuracy of 0.3 millimeters) mounted on an aluminum frame to track the ground truth position and orientation of the board and the test probes as shown in Figure 4.4. To facilitate this, we place IR retroreflective markers on both the boards and the test probes. The markers for the test probes are placed on 3D-printed crowns which are affixed to the top of the test probes, ensuring they remain in sight of the cameras as the hand grasps the probe. The motion

[7]OptiTrack Prime 17W

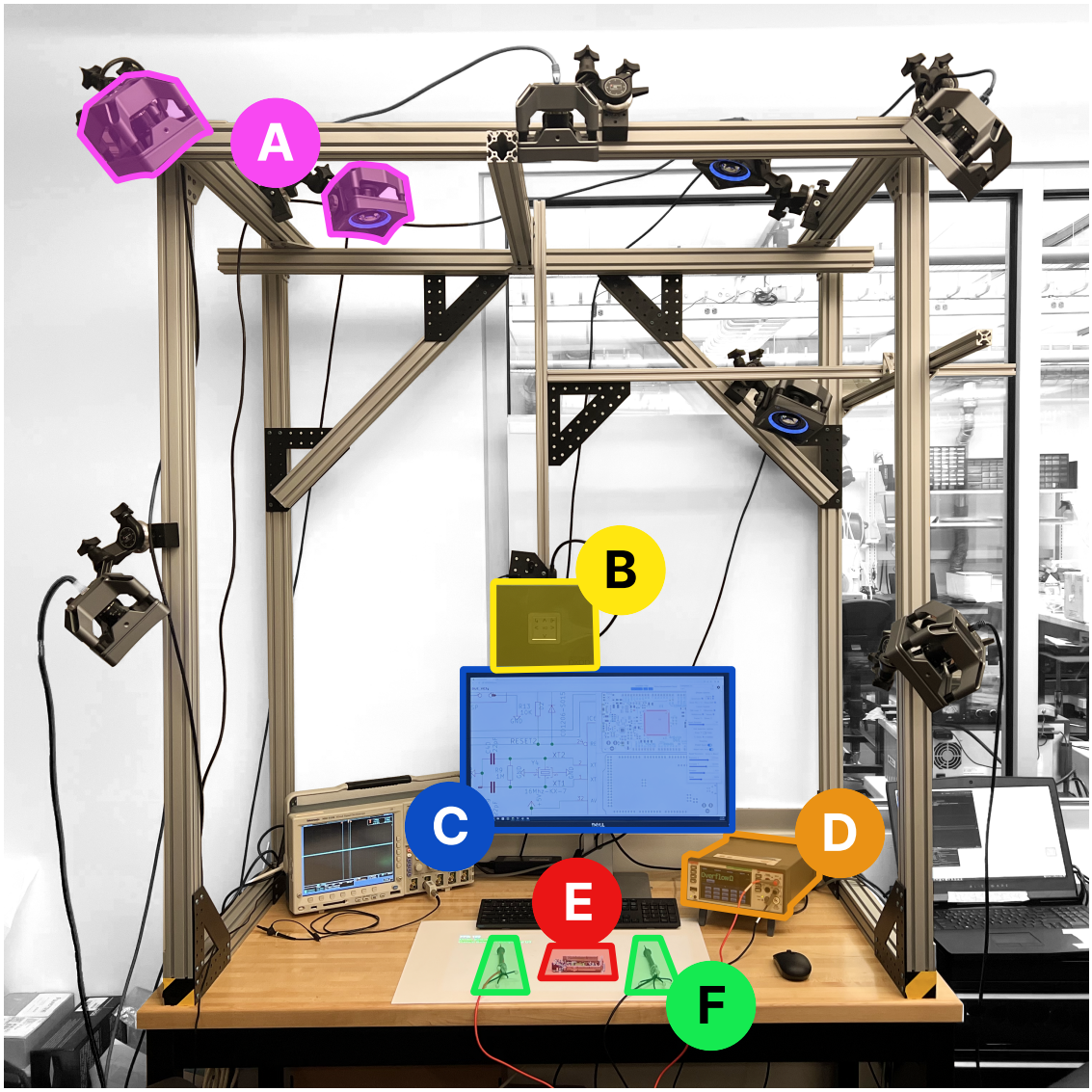


Figure 4.4: *The workbench. We used a motion capture system to track the pose of the board and test probes (A). A 1920-by-1080 resolution LED projector is used to project downwards on to the PCB (B). A screen in front of the user shows all the design files and settings (C). ARDW uses a connected benchtop digital multimeter to facilitate measurement (D). We use retroreflective markers to facilitate tracking of the board and test probes (E, F).*

capture system reports the pose of the board's and probe's crowns in the motion capture coordinate frame at 30 Hz. The data is piped over UDP[8] to the server backend. To address the issue of model fitting noise, which was especially significant when the probe's crowns were close together, we applied an exponential low-pass filter to both probe position (alpha=0.3) and board position (alpha=0.1) [171].

We use a one time calibration procedure to establish the position of the probe's tip relative to the crown's pose. To determine this, we press the probe tip against a surface to keep it fixed in space, while performing a circular motion with the probe. We use least squares to fit a sphere to the crown's poses and assign the sphere center to be the tip position.

Projector Display As debugging generally happens in a fixed bench-top location, we opted for projection-based AR. This allows for direct observation and manipulation of the board without an intervening display or screen, as in optical see-through augmented reality or mobile screen-overlay augmented reality, permitting for viewpoint independent augmentation simplifying render. We use a 1920-by-1080 resolution LED projector[9] to project downwards on to the PCB and white anti-static mat. The projector is mounted on an overhead aluminum frame approximately 0.6 meters over the surface, yielding a 48 by 27 cm display area as depicted in Figure 4.4. Positioning the board between the throw axis of the projector and the user generally avoids issues with users occluding the projection, except in cases where the user's hands are probing directly from above or the user places their head over the board.

We use a projector brightness of 1000 lumens, which was clearly visible in normal indoor office lighting. To establish the relationship between the tracking system coordinate frame and the projector's display, we project a 3-by-4 checkerboard and sequentially place the probe at each vertex, and use least squares to estimate the projector's projection transform. Since both our tracking system and board design files are in millimeters, we can scale the projection appropriately.

---

[8]User Datagram Protocol

[9]AXAA M7 Projector: `https://www.aaxatech.com/products/M7_pico_projector.html`

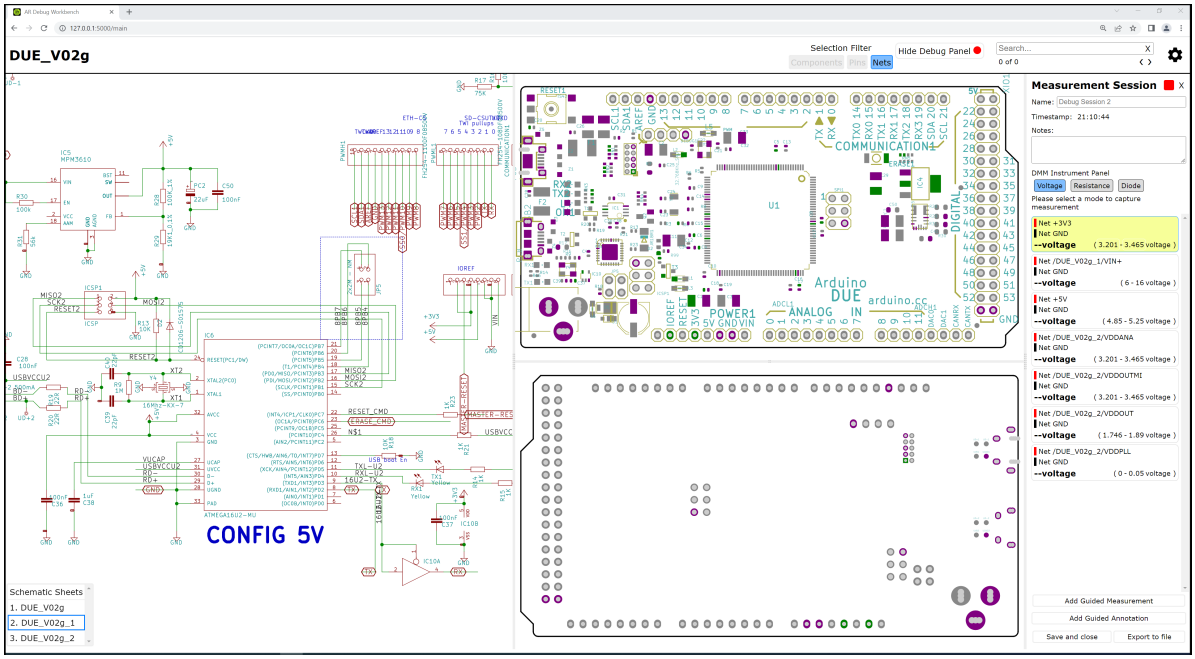


Figure 4.5: *A screenshot of the screen interface. Schematic is rendered in split screen with front and back layout views on the right. On the far right the measurement panel is open. In this capture, the system is in guided measurement mode with a set of tests loaded and the first measurement queued for capture. In the layout view, the system has highlighted positive probe points in green and negative probe points in purple (which is mirrored on the physical PCB). The selection filter for probe hitboxes (top right) has been limited to nets in this scenario.*

For the projected augmentations, the rendered content being projected is another front-end web page, similar to that of the screen interface, except that the only elements present are a black background and a multi-layer HTML5 canvas with the highlights. As with the screen interface, the projector display is connected via Socket.IO to the server backend to allow for real-time updates and facilitate remote collaboration. Augmentations of board elements, such as highlighting a specific pad, are generated in the same manner as the layout view of the screen interface, which has a 1:1 correspondence with the physical board when scaled correctly.

To facilitate alignment between PCB and the projected augmentations, we project all pads and edge cuts and allow the user to manually translate the board. If marker stickers are affixed to the board, we also provide a function to snap the projection to the board automatically, or to track the board in real time as it moves around. An outlier filter addresses instability when the probe and tracking markers are within the same vicinity or when a user's hand occludes the tracking markers from the cameras.

Connected Test Instrument To facilitate measurements, ARDWuses a connected benchtop digital multimeter (DMM)[10]. The server backend exposes DMM function selection via the screen interface's measurement panel, allowing the user to set the DMM mode (voltage, resistance, or diode), as well as mirroring the front panel's value. Mode selection and value querying is achieved with industry-standard SCPI commands via the VISA API over a USB 2.0 cable. By adhering to an industry standard command protocol, ARDW is easily extensible to accommodate additional DMM functions, different DMM models, or other types of test equipment, such as oscilloscopes, network analyzers, power supplies, and more.

Interaction with the Augmented Interface The augmented interface has two modes of interaction: selection and measurement. In selection mode, the positive probe of the multimeter serves as the selection probe. To make a selection, the user dwells on the

[10]Keithley DMM6500

component or pad with the tip of the probe, which was the method of selection preferred by electrical engineers in Augmented Silkscreen [28]. To detect when a probe's tip is dwelling, the backend keeps a short history of the tip position in the last 0.5 seconds and checks if all points lie within a 5 millimeter diameter sphere. Once a dwell has been detected, the current position of the probe tip is projected onto the 2D board layout and processed like a click in the screen interface. To help users see and account for tracking imprecision, each probe has a small colored dot projected at the calculated location of the probe tip. Additionally, all hitboxes are padded by 1 millimeter in all directions, allowing users to select the desired element even if the probe is not recognized as being directly on it.

As with the screen interface, there is often a need for disambiguation. There are two sources of selection ambiguity. First, hitboxes of different types generally overlap; for example, a selection on the pad of a component lies within the hitbox of the component, the pin, and the net of the pin. Second, the 1 millimeter hitbox padding means nearby hitboxes are likely to also be hit. To address the first source, the system includes a probe selection filter. From the screen interface, the user can choose between selecting components, pins, nets, or any combination of the three. However, in many cases, manual disambiguation is still necessary. When a selection is ambiguous, a disambiguation menu appears next to the board that lists the reference designators of the possible selections. The user can then make a selection within the menu by tilting their probe forward or back to manipulate the menu selection cursor, whose position corresponds to yaw of the probe, and dwelling the cursor within the desired menu item.

To avoid the issue of constant re-selection with minor adjustments in probe position, the server backend keeps track of a safe zone around the edges of the board and just above the highest component. Outside of this safe zone, no hitscan is necessary. Inside of the safe zone, the probe can make a selection, but it cannot make another one until it has been invalidated by leaving the safe zone. The result is an intuitive interaction: put the probe down on the board to make a selection, then lift the probe off the board and place it down again to make another.

Finally, unlike in the screen interface, deselection within the board is not supported. Instead, users can deselect by placing the probe tip down on the mat surface outside of the board. If the probe is merely set down, the probe tip rests several millimeters above the mat. To avoid a deselection in this case, the deselection zone extends only 1 millimeter above the mat.

In measurement mode, both the positive and negative multimeter probes are tracked and can select and disambiguate as in selection mode. When both probes have made a selection, a measurement is recorded from the connected test instrument and appears in the measurement sidebar of the screen interface. Unlike in selection mode, a probe is deselected as soon as it is invalidated by leaving the safe zone, rather than waiting for a deselection event, as selection in this mode is only for measuring specific elements.

Measurement mode also has additional disambiguation features. The selection filter works as before, except that 'component' is not available; an individual probe must select a pin or net. If a guided measurement has been specified in the screen interface, for example as part of a bring-up workflow, the system will try to automatically resolve ambiguous selections. If a probe makes an ambiguous selection that includes the expected pin or net from the guided measurement, the system assumes that the probe has been correctly placed and selects the expected element. However, if the expected element is not in the selection, the user will need to disambiguate as normal to proceed with a different measurement from the guided one.

4.4 Evaluation

To evaluate ARDW, we conducted a three-part user study with 10 electrical engineers. We derive the tasks within each part from common electrical engineering debugging workflows. In each section, we introduce new features of the system to the participant. In the design of the evaluation, we seek to strike a balance between structured usability testing with guided tasks and qualitative free-form exploration, to avoid the pitfalls premature usability testing can bring[50]. For qualitative feedback, we analyzed their responses via thematic analysis [20], first transcribing the interviews, then coding recurring themes, and finally noting outliers

from the norm. In part 1, we examine ARDW's effect on navigation between the board and design files. In part 2, we collect feedback on ARDW's use for PCB bring-up. In part 3, we gather participant's thoughts as they use ARDW in a set of free-form debugging tasks.

4.4.1 Participants and Procedure

We recruited 10 participants who hold electrical engineering roles in academic labs and/or industry. During our selection period, we confirmed that all of our participants regularly design and debug PCB designs. Participants' reported design focuses spanned from small, low power boards to FPGAs and complex mixed signal boards. A comprehensive list of their experience can be found in Table 4.1. Each study took between 60 and 70 minutes for a total of 11 hours of feedback, and participants were compensated for their time. Prior to the study, we ran a pilot participant to determine default settings to use such as hitbox size at appropriate study length. In the study, board tracking was not enabled.

Procedure of Task 1: Navigation We first introduced participants to the system by pointing out major interface components and demonstrating selection mode using the probe. To familiarize participants with selection mode, we presented a navigation task. Electrical engineers frequently navigate between their schematic, layout, and board during a debugging task[28]. Mirroring the evaluation in Augmented Silkscreen[28], Task 1 was split into two timed sub-tasks: (1a) Finding a component on the board given a target in the design files, and (1b) Finding a component in the design files given a target on the board. For Task 1a, a target component was highlighted in the design files of the Arduino Uno R3.[11] Participants selected the corresponding component on the Arduino board with their probe to indicate they found the component. A sound signalled success or failure. Participants could re-select if their first selection was incorrect. With augmented cross-linking enabled, the target component was highlighted on the board as well. In the control, only the schematic and layout were cross-linked as in standard ECAD tools. For Task 1b, a target component was highlighted

[11] `https://store-usa.arduino.cc/products/arduino-uno-rev3`

Table 4.1: *Recruited participant backgrounds and expertise.*

	Field	**Experience**	**Primary Tool**	**Designs**
P1	Industry	Design, Release, Functional check	Siemens PADS	Two-layer, large amplifier designs; High layer-count, high-speed PCBs such as PCIE and graphics cards
P2	Industry, Academia	Design, Release, Assembly, Functional check	Altium Designer, Eagle	Embedded firmware/hardware
P3	Industry	Design, Release, Functional check	Altium Designer, KiCAD	FPGA board; Embedded systems
P4	Industry, Hobby	Design, Release, Assembly, Functional check	KiCAD	Rework technician in industry; simple 2-layer sensing designs
P5	Academia	Design, Release, Functional check	KiCAD	Small, low power board for communication
P6	Industry	Design, Release, Engineering validation, Mass production	Altium Designer, KiCAD, Nexus	Small, high frequency signal board; Complex mixed signal
P7	Academia	Design, Release, Functional check	KiCAD	Small, single layer FPC for robotics
P8	Industry	Design, Release, Engineering validation, Mass production	Cadence, AutoCAD	FPC (2-3 layers, mixed signal); High-layer count, HDI motherboards with SoC, UFS, DDR memory; Power design
P9	Academia	Design, Release, Engineering validation	Altium Designer	Control board for robotic arm
P10	Academia	Design, Release, Assembly, Functional check	AutoCAD, DXF format	Sensing board for miniature robots

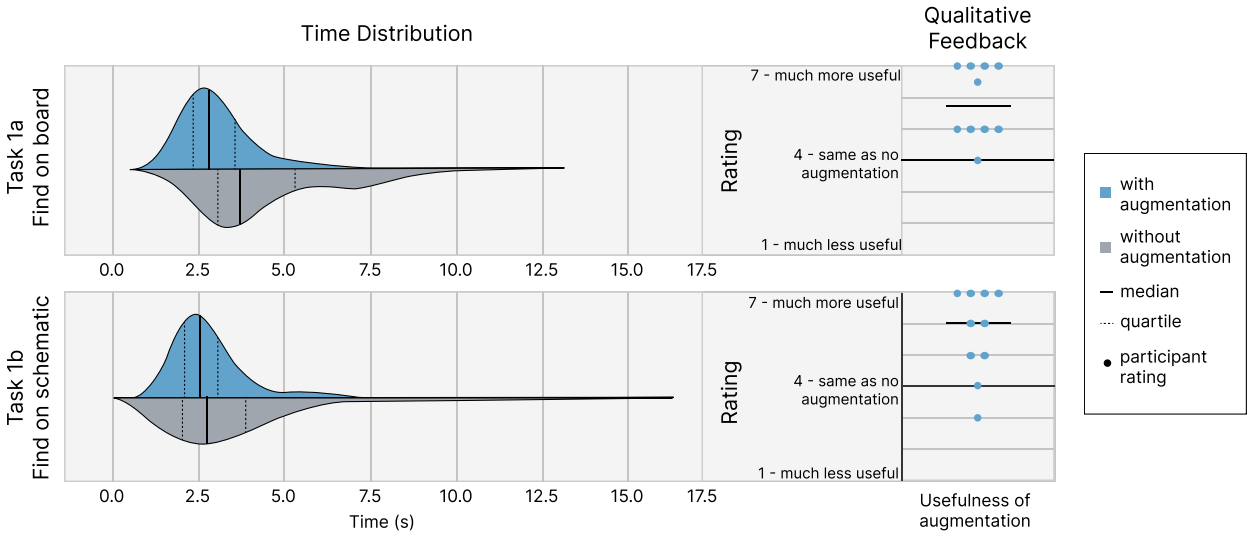


Figure 4.6: *The timing distribution and subjective scoring for Task 1a (Find on board) and Task 1b (Find on schematic). In Task 1a, participants performed faster with augmentation than without, while in Task 1b, participants performed similarly in both conditions.*

on the board and the participant selected the corresponding component in the schematic or layout. For each sub-task, participants were presented with 20 randomized component selections: 10 with augmented cross-linking, and 10 without augmented cross-linking. The order of presented conditions was counterbalanced across participants. The component selection filter was enabled across all tasks and conditions. In addition to timing the tasks, we collected qualitative feedback, first confirming whether the task featured in their own workflow and then using prompting questions such as "Would you find ARDW to be helpful, not helpful, or have no effect on your workflow?" and "What aspects did you like and not like about using the system for this task?" Finally, we recorded Likert scores for the question "For this task, how useful would ARDW be in your workflow?"

Procedure of Part 2: Bring Up Next, to introduce participants to measurement mode, the search bar, and multi-page schematics, participants performed a board bring-up task on

the Arduino Due[12]. We loaded a set of seven voltage rail measurements into the measurement panel. Participants first stepped through the measurements without the augmented cross-linking as a basis of comparison. Then we enabled probe tracking and board augmentation in measurement mode, and participants repeated the same measurement procedure. We recorded their general qualitative feedback, prompting them with the same questions as mentioned in Task 1, and then recorded Likert scores for the questions "For this task, how easy was the procedure with and without ARDW?" and "What was your confidence in executing the task with and without ARDW?"

Procedure of Part 3: Debugging Finally, we introduced the selection filter and demonstrated how to move between selection and measurement mode. Task 3 consisted of four free-form debugging tasks. We introduced errors into four different PCBs to represent a range of possible errors. For the Arduino Due, we soldered an incorrect feedback resistor in a buck network, causing the `+5V` rail to sit high (a fault they discovered while performing bring-up in Task 2). For the Arduino Uno, a diode near the power input was placed in reverse polarity. For the Sparkfun RedBoard[13], a mis-sized current-limiting resistor caused the power LED to be dimmer than expected. Finally, for the Sparkfun Sound Detector[14], an incorrectly biased op-amp network resulted in a clipped audio stream (as visualized on an adjacent oscilloscope). Each participant debugged two of the boards with ARDW and two without augmented cross-linking with order balanced via Latin square. Because of the unconstrained nature of this task, we pursued a qualitative coding approach. We encouraged participants to think aloud [88], and then asked a set of questions covering their overall impression of the system, limitations, wish-list items, and practical considerations.

---

[12] `https://store.arduino.cc/products/arduino-due`

[13] `https://www.sparkfun.com/products/13975`

[14] `https://www.sparkfun.com/products/12642`

4.4.2 Findings

Findings for Task 1: Navigation This initial task was the first time all participants interacted with the system. Participants' first impressions were that of excitement, especially after seeing the components first light up on the board in front of them.

The timing distribution and subjective scoring for Task 1a (*Find on Board*) and Task 1b (*Find in Design Files*) are presented in Figure 4.6. Shapiro-Wilk tests indicated all data was not normally distributed ($p<0.001$), so we used the Wilcoxon signed rank test to compare timings with and without augmentation. For Task 1a, participants performed faster with augmentation (Mdn=2.742) than without augmentation (Mdn=3.730). A Wilcoxon signed rank test indicated that this difference was statistically significant ($p<0.001$, effect size $r=1.537>0.5$, large effect). For Task 1b, participants had a similar performance with augmentation (Mdn=2.436) as without augmentation (Mdn=2.790). A Wilcoxon signed rank test indicated that there was no statistically significant difference ($p=0.228$, effect size $r=0.381>0.3$, medium effect).

Qualitative rating indicated generally positive sentiments about the usefulness of system in board navigation tasks with ARDW-enabled augmentations compared to the baseline case in both tasks. The high ratings were generally driven by ARDW's ability to reduce context-switching. Nine out of 10 participants mentioned that interacting with the projected augmentation reduced their mental effort to select or locate components, with six of the 10 participants mentioning the projection made it faster to select components. Additionally, a few participants mentioned that ARDW reduces the chances to make mistakes. Five of our 10 participants mentioned that ARDW's cross-linking functionality reduced the need to divide attention across multiple representations or tools. "It highlights things [that I'm looking for] and I know where to go, so it really reduces the amount of time that I need to go back and forth, and dealing with a 3rd item [the mouse]" (P2, Q43). "Like, it would be nice to never have to pull up a board [layout] file, and then only click on a schematic and then, bam!, it shows me where it is on the board. You could eliminate half of the screen. That

would be great" (P8, Q44).

Participants generally felt that the larger and denser the board, the more useful the cross-navigation representation would be. "This would be really helpful, I can see that for larger boards this is more helpful, and for smaller boards with smaller components it is extremely helpful. I have had cases with 0603 resistors, and in those cases, it would really help me to find those components" (P2, Q45). For larger, distinctive components the benefit of augmentation is less realized. "I'll use it all the time for smaller components, but for larger components, by eyes is faster" (P1, Q46).

All participants picked up the probe selection mechanism quickly in the practice rounds for Task 1. However, in observing the selection behaviors of our participants, we noticed a split between participants touching down on the component versus hovering a few millimeters above components to select. While for some, contact with the component confirmed they were selecting an item, others found it could sometimes slightly shift the board resulting in augmentation misalignment. For example, one participant mentioned "[from] my understanding you kind of want to not necessarily tip off the thing [and shift the board], but you want [the projected dot visible under the probe tip] to be on component... you don't really have to touch it, right?" (P4, Q47). The board shifting from the augmented projection was the main source frustration expressed by nearly all participants. This caused their selection to be less precise, slowing their ability to get into the component's hitbox with the probe and decreasing their confidence in the system. "The only pain in the butt is just applying pressure to the board, the board rocks, so you're naturally going to get misaligned" (P8, Q48). "Due to the not perfect alignment... I'm definitely towards less confident [because] I have to have the same double check as without augmentation" (P2, Q49). This was most evident for participant P3. A tracking camera had been bumped prior to their study, resulting in poor alignment between the probe's reported and actual position. The probe tracking's imprecision made tasks extremely tedious, causing frequent mis-selections (resulting in the lowest data point in both qualitative rating scales).

On the other hand, a number of participants appreciated the directness of interaction

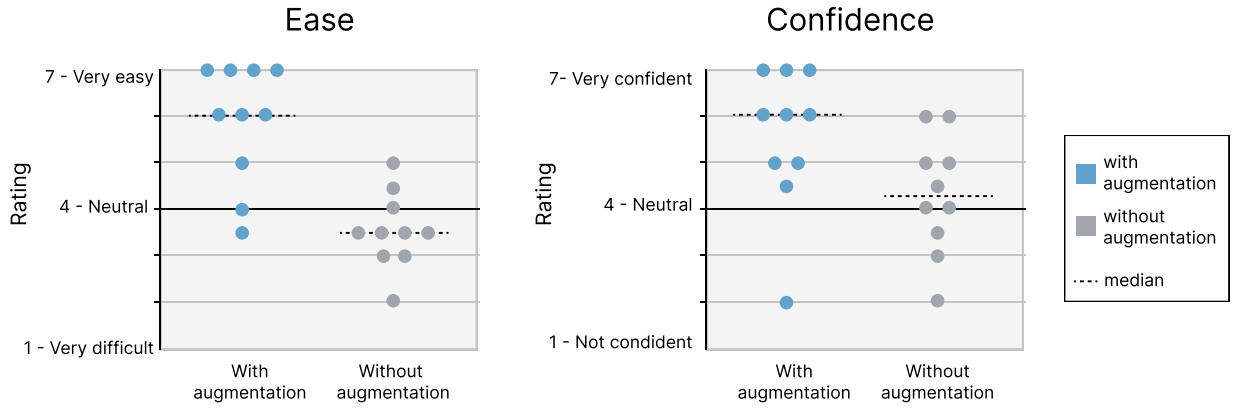


Figure 4.7: *Participant rating from task 2 for ease and confidence.*

afforded by selecting with the probe. "I can go there [tap on the component] directly and get the link to the data sheet" (P2, Q50).

Findings for Task 2: Bring Up Many of the same findings that we learned from Task 1 (Navigation) held true for this bring-up task as well.

In the bring-up scenario, the benefit of reduced context switching was even more noticeable. In the condition without augmentation, participants would typically have to reference the list of required measurements at least once and the design files at least twice for every single measurement to find appropriate probe points for their positive and negative probes. When a probe point proved too small or difficult to access, this back-and-forth multiplied as participants returned to the design files to identify another suitable point. With augmentation, participants' area of interaction consistently remained on the board. While we did not explicitly record times for the task, as we asked participants to think aloud, they agreed that they were much faster with the augmented measurement. Participants generally found augmentation to make bring-up easier and were more confident in performing bring-up (see Fig. 4.7). "[It's] much better with the AR assistant because one, it highlights where the net is with the color code, you don't need to find net then pin, and two, voltages are automatically loaded into

the spread sheet" (P1, Q51).

However, all participants were again challenged by imprecision in probing due to movement of the PCB, exacerbated by three factors: participants used both the positive and negative probes to take measurements, applying force ensure good contact; the board was connected to power via its DC barrel jack and the wire could tug on the board; and the pins on the Arduino Due were a relatively smaller target than the components on the simpler Arduino Uno.

The smaller augmentation targets also created additional challenges on the precision of the projected highlight as well. Four out of 10 participants also mentioned that the highlight was not precise enough and that the highlights on multiple adjacent fine pitch pins blurred together, therefore reducing their confidence of what to probe for the corresponding measurement. "But yeah, definitely concerned with resolution, because I think most of the stuff that we do is either just denser or smaller" (P8, Q52). "This last one was so small it highlighted both of them" (P4, Q53). One participant mentioned that they would still prefer using a system that is slightly mis-calibrated over having to constantly switch between using probes and mouse. "The augmentation made it easier to probe, [but] when you are probing it, it does shift away. And [the projection] doesn't move with it. That's kind of annoying, but that's still better than having to remove your hands from the probe to find it. So I think it is better in terms of efficiency" (P5, Q54). Additionally, because of the number of targets highlighted in the bring up task, especially with all the pins associated with the `GND` net, four participants indicated the board was overpopulated with highlights, making the probe points more difficult to discern because of the relative brightness and busyness of the projections. "It's mostly just because there's so many things being highlighted. Versus [when] I'm just trying to look for a component, it's just highlighting one thing... dealing with smaller things it gets a little busy" (P2, Q55). "It messes with how I see it" (P9, Q56). These participants indicated that they would like the option to turn off the `GND` highlight since they would often solder a separate lead or have a dedicated test point. Following the study, we added a toggle to turn off the highlight for `GND`.

While most participants indicated that this feature would be useful in their own workflow, it is also worth noting that four of the 10 participants mentioned that this bring-up feature (and more generally the pre-specified measurement feature) is not useful because they do not perform bring-up tasks in a fixed sequence, instead performing their net selection on-the-fly. The split in opinion was divided by complexity of the design they usually worked with: these participants generally focused on miniature low-net count designs, while two other participants who work in industry on more complex designs acknowledged the benefits of having the measurements automatically recorded.

Findings for Task 3: Free Form Debugging The process of debugging again required the participants to cross-reference between the layout/schematic and the PCB. We observed that the findings from the two previous tasks are a common thread through Task 3 as well. Notably, the benefits of ARDW in highlighting when locating the components were again resonated from all 10 participants in this task. Of which, five of them mentioned that it reduced the amount of effort required. For the Sound Detector board, the power cord caused the typical orientaiton of the board on the table to rotated from that of the layout. Three participants specifically mentioned that the augmentation was particularly helpful in this case. "Augmentation [was] helpful...like a split second improvement because [the layout view] is flipped from the way this [sound detector] is wired up. Like I had to do a little bit of mental gymnastics... especially because, again, it's a board without silkscreen labels, so yeah, the augmentation helped for that" (P6, Q57). "Here's a much more compelling case because it's not laid out in the right direction, so I'm doing these transformations in my head to be like, which side of it is that on? And the [augmentation] would answer that immediately for me without any room for error, so that's cool" (P3, Q58).

The movement of the PCB which resulted in imprecision in probing was again echoed by almost all participants. Some of participants taped down the PCBs on to the mat, which resulted in a smoother experience: "If there was maybe, along with the augmentation setup, some kind of way to stick the board to one spot, that would be more practical" (P7, Q59).

Participants' level of expertise and field of work influenced their perspective on how useful this system is for them. Most participants agreed that the feature to locate components using the augmented projection is helpful, especially for larger and/or denser boards with smaller components. However, participants' definition of what is a dense or large board depended on their current level of experience. It was generally agreed that this system is most particularly useful when they are interacting with new PCBs. "It makes it so effortless... it's not like it takes that much effort to find something, but it reduces it to like, absolute zero. " (P9, Q60). "This is definitely a good tool for if you're debugging a PCB you're not familiar with" (P6, Q61).

We noticed that only a small amount of participants used the disambiguation menu during their debugging process. Initially, participants would use the menu to differentiate between pins and components, but as the debugging process continued, the participants primarily used the selection filter.

4.5 Discussion

Through our evaluation, we found common themes around the strengths and limitations of our system, which we summarize in Table 4.2. Through our feedback session, we extract design considerations for future systems where we emphasize them in **bold** throughout the rest of this section.

4.5.1 Strengths of ARDW

Reduced context-switching Across all tasks, participants noted that **reducing context-switching, specifically the need to look between design files and the board, made their debugging experience easier, faster, and more confident**. Despite the imprecision of the system at small scales (see next section for further discussion), the benefit of augmentations for the most part outweighed the challenges. Participants consistently rated the augmentation more highly than without augmentation in navigation and measurement tasks, with one commenting that the ability of the system to "get them in the general

Strengths	Limitations
• Reduced context switching • Directness of probe interaction • Screen Interface Usability	• Imprecision of augmented highlights at fine scale • Imprecision of probe localization at fine scale

Table 4.2: *The high-level strengths and limitations of ARDW*

vicinity" (P9, Q62) was already of some help.

Interestingly, even if the system did not make participants faster in certain tasks (e.g. Task 1b), participants felt that the system made them more efficient. We believe this is because of the lower cognitive load the system fostered even for simple tasks. As P9 put it, "I think that it would be easier for augmentation because like, for example, when I was looking for this right here... it just feels like I'm using no brain power at all... It's like the difference between zero [effort] and like point one [effort]. But like those point ones add up over the course of hours. So I can see it being really, really useful for a long term thing."

Finally, using ARDW gave participants greater confidence when identifying components and probe points or when approaching a new board. P10, who mentioned that "ease wise, and confidence... I think both are directly correlated", gave a Likert score of 7 for both the questions on ease and confidence when using ARDW in Task 2: Bring Up.

Four out of 10 participants sought to further reduce context switching by integrating the system with automated hypothesis generation and guidance. "I almost wanted it to tell you what pin it is for me [instead of having to] look up and [specify] which pin on my program" (P8, Q63). "If [the system] can say hey, check this out, it's lower than it's supposed to be, that would be helpful, kind of warns you. Or you could go for the diode and [it might

say], hey, is your [diode in] reverse? So that'd be helpful... helps you check against the schematic" (P4, Q64). Taking this thread further, future work could allow for ARDW to assist in physical computing environments that consist of both code and embedded hardware portions. As users step through code, sub-circuits related to that code could be augmented directly on the board. Paired with the ability to code step, this could make a compelling way to more directly pinpoint errors that cross code and hardware.

Directness of Probe Interaction Participants appreciated the use of a probe as interactive tool, commenting that the directness of interaction on the board made for a more seamless navigation experience across representations. "It'd be really useful to just like, instead of looking back and forth trying to find a component, it'd be really easy to just click on it and it highlights [on the schematic] so I don't have to search in the schematic because... often I use schematics way bigger than [Arduino Uno] and [it's] really time consuming" (P4, Q65). "I definitely like the backwards from board into schematic better, because then if I'm looking something where the heck is this, and then I just [find] it right there. And that's really useful" (P6, Q66).

Some suggested building out the augmented interface to allow for greater interactivity on the anti-static mat itself. **The user's attention should generally remain at the point of interest, in this case the mat where the PCB is placed.** Through observation, users' attention typically jumps between the schematic or layout view, the PCB, the digital multi-meter, and the measurement panel. By making use of the real estate that the mat has, more relevant information can be provided to the user, without them having to take their attention away from the PCB. P5 said: "My top dislike is having my attention in two different places. Or if the tension is split, like fifty-fifty and there's more movement." One participant expressed the desire for measurement information to be shown right next to the PCB. Another participant expressed that they would love to have an API to generate interactive graphics directly on the mat itself, so they could quickly change test modes by simply tapping on the mat. Future work could explore making the mat surface a more interactive element of the

system.

Screen Interface Apart from the augmented cross-probing, a few participants commented that they appreciated the utility of screen interface alone. They indicated that the cleanliness and smoothness of the interface (which is comparatively lightweight to the busyness of a typical ECAD tool) enabled a better experience. They also indicated that the interface encouraged them to cross-probe between schematic and layout, which was helpful for debugging, something that traditional ECAD programs do not directly emphasize.

4.5.2 Limitations of ARDW and Future Work

Precision and Clarity of Projected Augmentations While users expressed that the greatest benefit for such a system would be for large densely-packed boards, dense boards created the greatest challenge to augment precisely. **While some users appreciate augmentations that can direct to a general target area, greater precision in augmentation would unlock finer targets further reducing context switching.** While the highlights generally worked well for larger components and targets (0603 packages and up), smaller targets such as 0201 package components or dense IC pins proved challenging to highlight effectively due to imprecision of the highlight. While the projected pixel size is approximately 0.25 mm by 0.25 mm across the 43 cm by 27 cm area, the ability of our relatively budget projector's optics to resolve those pixels was limited. Furthermore, users did not benefit from the large projected area as they debugging happened within one location in the mat. Future systems would benefit from concentrating resolution in a tighter space and from better-resolving projection optics to maintain sharpness. Future work could benefit from a focus-free projector such as a laser scanning projector which could help to maintain sharpness across the component z-heights. This could also allow for in-focus augmentation even if the board is held in mid-air.

While most augmentations were easily visible especially for larger components or diffusive IC packages, some highly specular components, like polished metal housings or glossy solder-

mask, resulted in changing augmentation clarity from different viewpoints. Additionally, the system is extremely sensitive to perturbations. A combination of the workbench being on wheels and the weighty projector being mounted on a cantilevered arm causes the projected augmentations to wobble a couple millimeters when the table is bumped. This made using the system a more delicate procedure. For future systems, ensuring rigidity of the projector mounting structure or high speed correction of projection would make the augmentation alignment more tolerant to vibration.

To our surprise, only three participants brought up blocking the projection as they looked to take a measurement. Generally, users approached probe targets from an angle to avoid occlusions. However, when the probe point required a vertical approach, participants expressed reservations: "Sometimes, like, you got to really come in... basically vertically and the way this is working from above kind of restricts that motion" (P6, Q67). The fact that the board sat between the user and the projector's throw axis helped avoid occlusion to some extent since the light came from an angle beyond where the user's body typically was rather than directly straight down on the board. However this introduced a new issue, where taller components' projections were translated because the projection arrived off-axis. In future systems, this could be remediated by performing distortion correction relying on a 3-D model of the PCB or depth map to apply the correct shift for tall components.

Precision of Probe Tracking Another major challenge for the usage of the system is the precision of the tracked probes in relation to the board. We find that **good alignment and calibration of tracked probes is crucial for users to want to use the system.** In a number of instances, especially for smaller targets users would need to hunt for the hitbox or use probe cursor dot to adjust the position of the probe to achieve their desired selection. There were two sources of imprecision: (1) the physical board moved in relation to the virtual board due to bumping the board, and (2) the inherent system inaccuracy of the probe point due to mis-calibration of the tracking or the transform between projection and tracking.

To address these challenges, participants suggested a few methods. The simplest was

to fix the board in place with an adjustable vice or putting the board on standoffs (some participants naturally did this by taping down the sound detector board in Task 3). Future work should explore this avenue as a low-hanging fruit to alleviate probing issues.

Another method would be to continually track and update the board position. We implemented continuous board tracking by applying retroreflective stickers to the board surface. Translational augmentation registration accuracy (combining board tracking accuracy and projection accuracy) is ±0.4mm MAE as implemented. At this accuracy, qualitatively, 0402 pads and 0201 packages are accurate, with 0201 pads and dense IC pins (e.g. on QFN packages) being close but imperfect (e.g. see 1:15 of demo video). For the user study, board tracking was not enabled as our pilot participants probed the board such that it did not tend to move (applying little lateral force on the PCB with friction from ESD mat). This turned out to be the exception amongst participants in the user study. We brought back two participants for an informal, post-hoc survey to compare their experience with board tracking enabled. Both participants provided strongly positive feedback indicating that it allowed for both more precise selections but also a more ergonomic and comfortable debugging experience, but confirmed it fell short of selecting the smallest targets such as IC pins. "So overall I definitely feel this is a lot more helpful, just because I trust here what it generate more than I trust myself when lining it up." (P2, Q68) Future work could look into having the board sit on a tray with markers or placing the board markers on a crown connected to the board.

We found that users generally did not engage with the projected disambiguation menu list because of the additional friction in brought in trying to select a targets in a dense area. The list contained a set of textual names, but would be likely more effective with a spatial selection scheme to avoid the users from having to think about the type of element they're looking to select. "Instead of just giving me the net names of the two pads, show me an enlarged display of the region under the probe and let me select the pad more visually. Having net names on this is still nice though" (P8, Q69). That being said, it is likely that better precision that can do away with the disambiguation menu entirely would be ideal.

Other considerations **A lightweight set up process is required for electrical engineers to choose to employ this system over their current workflow.** The value the system brings during work must significantly exceed the effort to set it up. Regardless of whether the participant works at a fixed workbench or needs the system to be portable, minimum set up effort is required for users to choose to use ARDW. These factors include the physical set up, software set up (calibration of the projector, probes, and tracking cameras), and setting up the design files. "If it could be movable from bench to bench...without much tweaking, that would make it way more practical and usable. Because I think anytime I have to grab something and be like, oh man, I've got to calibrate this thing, it just hinders me wanting to use it at all" (P8, Q70).

System cost is 33,390 USD, mainly driven by motion capture rig price (Optitrack cameras and equipment: 32k USD, projector: 700 USD, monitor: 300 USD, probes: 40 USD, desk: 200 USD, cage: 150 USD). From a cost viewpoint, participants agreed that the system should not significantly exceed the price of other professional-grade test bench equipment such as bench top DMMs and oscilloscopes (2.5k USD to 35k USD). Shedding the motion capture system for a cheaper RGB camera solution can help to make the system more practical. We are particularly encouraged by the recent work in PCB component segmentation via deep learning [119], and works exploring precise probe tracking with RGB cameras [175].

Lastly, while we did not get to test the remote collaboration capability due to the long study length, participant gave feedback that the the system would be beneficial for remote collaboration, especially with people who are new to the board or for non-electrical engineers who have to perform basic debugging. "We have engineers [in another office] but none of them are electrical engineers... but they deal with hardware. So they'll get a board, they'll boot it up, do some stuff and find out, stop working. But that's the time where I need to help them debug over the phone. And they don't have the technical know how to read a schematic where this would be wildly beneficial, if someone could just show them like where it was on this board. So I don't have to pull up the assembly drawing myself or go through that process" (P8, Q71).

Study Limitations Participants generally agreed that the tasks provided within the study were realistic and representative of the items they address in their own work. However, for users who focused on smaller, simpler boards, they found that bring-up task to be less applicable to their own workflow. For Task 3 (Debugging), we used the same tasks across all users. Therefore, the difficulty of a given task differed based on the participant's own experience and concentration area. For this task, we collected only qualitative feedback due to unconstrained nature, but it would have been interesting to collect qualitative rating scores to (1) understand the difference in perceived benefit across differing experience levels and (2) quantify the magnitude of perceived benefit against the other tasks which were more directed. Finally, because of the length of the study, participants did not directly engage in a collaboration task. Future work should explore this area further.

4.6 Conclusion

In this chapter, we build and test ARDW, a workbench that leverages projected augmented reality and tracked probes to assist electrical engineers in debugging PCBs. By enabling the ability to select and highlight elements across the schematic, layout, and physical board, ARDW reduces context-switching between board representations, allowing for users to more efficiently localize items on the PCB and capture measurements. Users appreciated the directness of interaction afforded the tracked selection and measurement probes and the performance of our in-browser application. We provide design considerations and recommend paths of inquiry for future systems, including techniques to minimize imprecision of augmentation and to explore remote collaboration scenarios.

Users are excited to see further development in this area. When we mentioned compensation logistics after our study, a number of participants expressed that they signed up for the study just because it "sounded cool", with one participant exclaiming "man, I wish to have something like this in my workplace to be honest!" By sharing our findings and open sourcing our code, we hope this work inspires further research into tools to support hobbyists and industry professionals alike.

Chapter 5

CLEARBUDS

This chapter describes our wireless earbud system, ClearBuds, that uses both spatial and acoustic cues to perform real-time speech enhancement. This technique enables users to be able to take calls on-the-go even in very noisy environments. This chapter was published at ACM MobiSys 2022 [29]. An illustrative video figure can be accessed here: `https://youtu.be/AVtFa0dN4mQ`.

5.1 Introduction

With the rapid proliferation of wireless earbuds (100 million AirPods sold in 2020 [120]), more people than ever are taking calls on-the-go. While these systems offer unprecedented convenience, their mobility raises an important technical challenge: environmental noise (e.g., street sounds, people talking) can interfere and make it harder to understand the speaker. We therefore seek to enhance the speaker's voice and suppress background sounds using speech captured across the two earbuds.

Source separation of acoustic signals is a long-standing problem where the conventional approach for decades has been to perform beamforming using multiple microphones. Signal processing-based beamformers that are computationally lightweight can encode the spatial information but do not effectively capture acoustic cues [158, 83, 33]. Recent work has shown that deep neural networks can encode both spatial and acoustic information and hence can achieve superior source separation with gains of up to 9 dB over signal processing baselines [148, 96]. However, these neural networks are computationally expensive. None of the existing binaural (i.e., using two microphones) neural networks can meet the end-to-end latency required for telephony applications or have been evaluated with real earbud data.

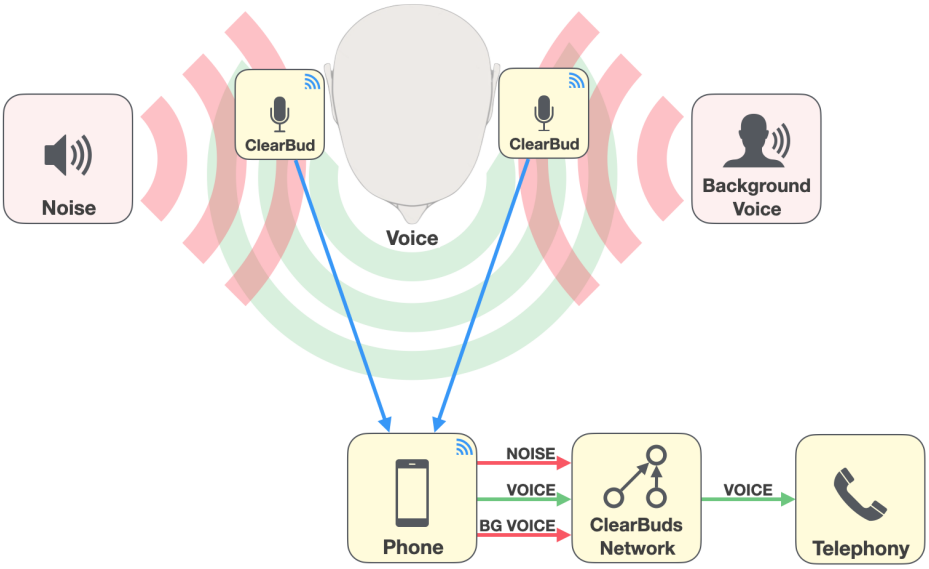


Figure 5.1: *ClearBuds Application. Our goal is to isolate a user's voice from background noise (e.g., street sounds or other people talking) by performing source separation using a pair of custom designed, synchronized, wireless earbuds.*

Commercial end-to-end systems, like Krisp [84], use neural networks on a cloud server for single-channel speech enhancement, with implications to cost and privacy.

We present the first mobile system that uses neural networks to achieve real-time speech enhancement from binaural wireless earbuds. Our key insight is to treat wireless earbuds as a binaural microphone array, and exploit the specific geometry – two well-separated microphones behind a proximal source – to devise a specialized neural network for high quality speaker separation. In contrast to using multiple microphones on the same earbud to perform beamforming, as is common in Apple AirPods [6] and other hearing aids, we use microphones across the left and right earbuds, increasing the distance between the two microphones and thus the spatial resolution.

To realize this vision, we need to address three key technical challenges to deliver a functioning, practical system:

1. Today's wireless earbuds only support one channel of microphone up-link to the phone.

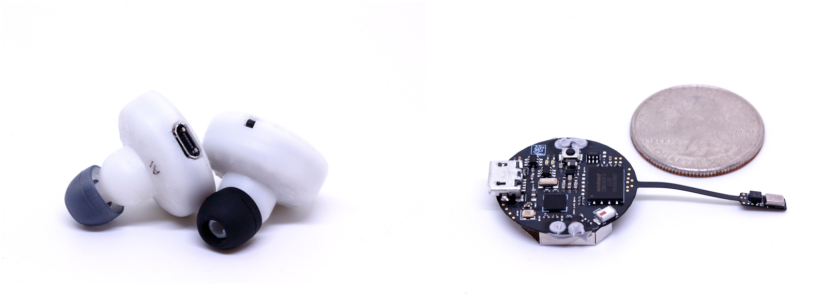


Figure 5.2: *ClearBuds hardware inside 3D-printed enclosure and when placed beside a quarter.*

AirPods and similar devices upload microphone output from only a single earbud at a time. To achieve binaural speaker separation, we need to design and build novel earbud hardware that can synchronously transmit audio data from both the earbuds, and maintain tight synchronization over long periods of time.

2. Binaural speech enhancement networks have high computational requirements, and have not been demonstrated on mobile devices with data from wireless earbuds. Reducing the network size naively often leads to unpleasant artifacts. Thus, we also need to optimize the neural networks to run in real-time on smart devices that have a limited computational capability compared to cloud GPUs. Further, we need to meet the end-to-end latency requirements for telephony applications and ensure that the resulting audio output has a high quality from a user experience perspective.

3. Prior binaural speech enhancement networks are trained and tested on synthetic data and have not been shown to generalize to real data. Building an end-to-end system however requires a network that generalizes to in-the-wild use.

To achieve this system, we make three technical contributions spanning earable hardware and neural networks.

• **Synchronized binaural earables.** We designed a binaural wireless earbud system (Fig. 5.2) capable of streaming two time-synchronized microphone audio streams to a mobile

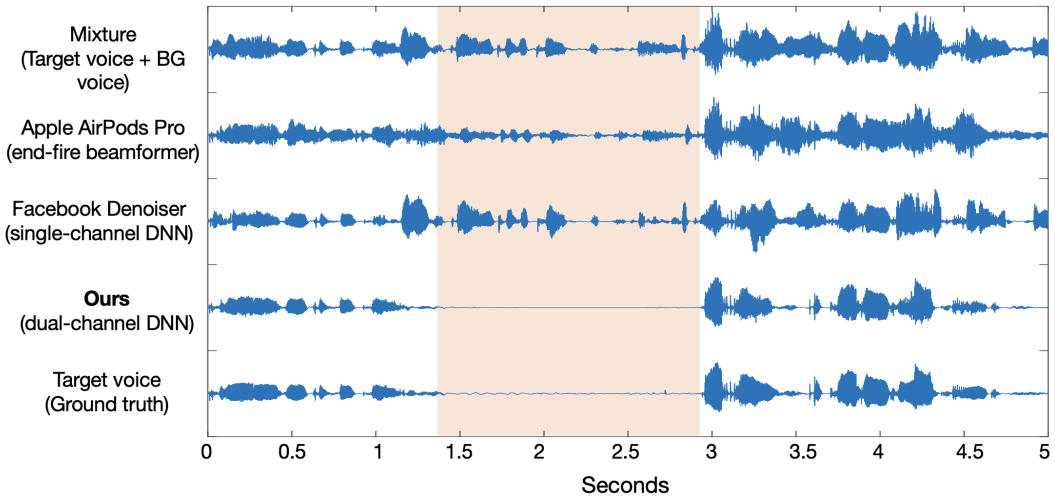


Figure 5.3: *Background voice performance. We use spatial cues to separate background voices from the target speaker, even when the background voice is louder than the target voice. This is evident when the target speaker is silent but background voice continues to talk (highlighted in orange). Apple AirPods Pro uses an endfire beamformer to partially suppress background voice. The mono-channel Facebook Denoiser (Demucs) is unable to suppress the background voice. Clearbud's network removes the background voice, approaching ground truth.*

device. This is one of the first systems of its kind, and we expect our open-source earbud hardware and firmware to be of wider interest as a research and development platform. Existing earable platforms such as eSense [69] do not support time-synchronized audio transmission from two earbuds to a mobile device. We designed our DIY hardware using open source eCAD software, outsourced fabrication and assembly ($2K for 50 units), and 3D printed the enclosures.

• **Lightweight cascaded neural network.** We introduce a lightweight neural network that utilizes binaural input from wearable earbuds to isolate the target speaker. To achieve real-time operation, we start with the Conv-TasNet source separation network [96] and redesign the network to achieve a 90% re-use of the computed network activations from the previous time step for each new audio segment (see 5.2.2). While these optimizations make

this network real-time, they also introduce artifacts in the audio output (i.e., crackling, static). Interestingly, these artifacts have little effect on traditional metrics, like Signal-to-Distortion Ratio (SDR), but have a noticeable effect on subjective listening scores (see 5.4.2). These artifacts however are often visible in a frequency representation of the audio. To address this, we combine our mobile temporal model with a real-time spectrogram-based frequency masking neural network. We show that by combining the two networks and creating a lightweight cascaded network, we can reduce artifacts and improve the audio quality further.

• **Network training for in-the-wild generalization.** Training the network in a supervised way requires clean ground truth speech samples as training targets. This is difficult to obtain in fully natural settings since the ground truth speech is corrupted with background noise and voices. Training a network that generalizes to in-the-wild scenarios also requires the training data to mimic the dynamics of real speech as closely as possible. This includes reverb, voice resonance, and microphone response. Synthetically rendered spatial data is the easiest type of data to obtain, but most different from real recordings, while real speakers wearing the headset in an anechoic chamber provide the best ground-truth training targets, but are the most costly to obtain. Synthetic data can simulate various reverb and multi-path that are not captured in an anechoic chamber. Our training methodology uses large amounts of synthetic data simulated in software, small amounts of hardware data with speakers embedded into a foam mannequin head and small amounts of data from human speakers wearing the earbuds in an anechoic chamber (see 5.3) to create a neural network that generalizes to users and multi-path environments not in the training data.

We combine our wireless earbuds and neural network to create ClearBuds, an end-to-end system capable of (1) source separation for the intended speaker in noisy environments, (2) attenuation and/or elimination of both background noises and external human voices, and (3) real-time, on-device processing on a commodity mobile phone paired to the two earbuds. Our results show that:

• Our binaural wireless earbuds can stream audio to a phone with a synchronization error

less than 64μs and operate continuously on a coin cell battery for 40 hours.

• Our system outperforms Apple AirPods Pro by 5.23, 8.61, and 6.94 dB for the tasks of separating the target voice from background noise, background voices, and a combination of background noise and voices respectively.

• Our network has a runtime of 21.4ms on iPhone 12, and the entire ClearBuds system operates in real-time with an end-to-end latency of 109ms. For telephony applications, an ear-to-mouth latency of less than 200ms is required for a good user experience [62].

• In-the-wild evaluation with eight users in various indoor and outdoor scenarios shows that our system generalizes to previously unseen participants and multipath environments, that are not in the training data.

• In a user study with 37 participants who spent over 15.4 hours and rated a total of 1041 in-the-wild audio samples, our cascaded network achieved a higher mean opinion score and noise suppression than both the input speech as well as a lightweight Conv-TasNet.

We believe that this work bridges state-of-the-art deep learning for blind audio source separation and in-ear mobile systems. The ability to perform background noise suppression and speech separation could positively impact millions of people who use earbuds to take calls on-the-go. By open-sourcing the hardware and collected datasets, our work may help kickstart future research among mobile system and machine learning researchers to design algorithms around wireless earbud data.

5.2 ClearBuds Design

We first introduce our lightweight neural network architecture. We then describe system design of our hardware platform and our synchronization algorithm.

5.2.1 Problem Formulation

Suppose we have a 2 channel microphone array with one microphone on each ear of the wearer. The target voice is speaking with a signal $s_0 \in \mathbb{R}^{2\times T}$ in the presence of some background

noise **bg** or other non-target speakers $s_{1..N}$. There may also be multi-path reflections and reverberations **r** which we would also like to reduce, i.e., $\mathbf{x} = \sum_{i=0}^{N} \mathbf{s}_i + \mathbf{bg} + \mathbf{r}$. Our goal is then to recover the target speaker's signal, s_0, while ignoring the background, reverbations, or other speakers. We also must do so in a real-time way, meaning that the a mixture sample $\mathbf{x}_t$ received at time t must be processed and outputted by the network before $t + \mathbf{L}$ for some defined latency **L**. We refer to the non-target speakers as "background voices". These background voices may be at any location in the scene, including very close to the target speaker and their angle can change with time and motion.

5.2.2 Neural Network Architecture Motivation

Our network needs to perform in real-time on a mobile device with minimal latency. This is challenging for several reasons. First, the processing device has a much lower compute capacity, especially compared to cloud GPUs. Additionally, the network should separate non-speech noises as well as unwanted speech. To do this, it must learn spatial cues and human voice characteristics. Finally, the resulting output should maximize the quality from a human experience perspective while minimizing any artifacts the network might introduce.

Our network, which we call *ClearBuds-Net* or *CB-Net*, is a cascaded model that operates in both time and frequency domains. The full network architecture is illustrated in Fig. 5.4 and contains two main sub-components: A dual-channel time domain network called *CB-Conv-TasNet*, and a frequency based network called *CB-UNet*.

CB-Conv-TasNet The first component of separation method is a time domain network that is based on a multi-channel extension of Conv-TasNet [96]. This is a network in the waveform domain that has a Temporal Convolution Network (TCN) structure, lending itself to a causal implementation with intermediate layer caching [117]. We use depthwise separable convolutions [59] to further reduce the number of parameters and make the design real-time. We call this network CB-Conv-TasNet since it is an optimized version of the original Conv-TasNet.

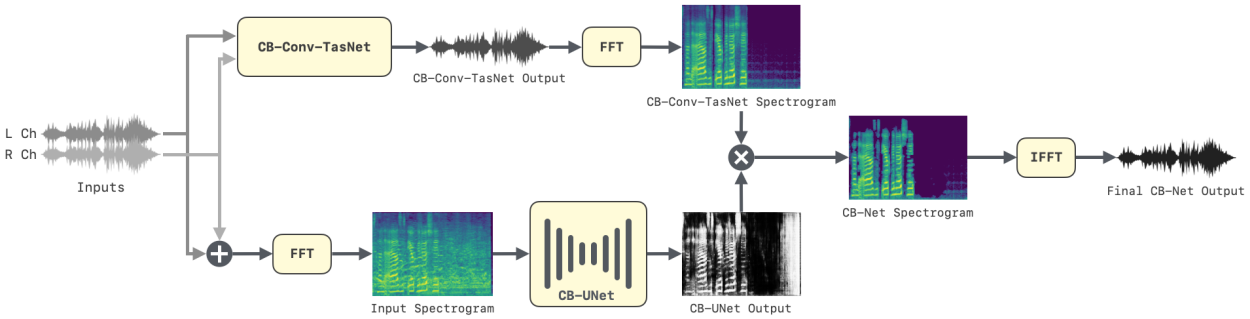


Figure 5.4: *Network Diagram of CB-Net. Our network contains a time-domain component, shown in the top as CB-Conv-TasNet, and a frequency domain component, shown on the bottom by CB-UNet.*

A key feature of the time domain approach is that it can easily capture spatial cues in the network. In our application, the desired source is always physically between two microphones, thus the voice signal will reach the microphones roughly at the same time. In contrast, background or other speakers are typically not temporally aligned and will reach one microphone earlier or later. By feeding two time synchronized channels into the neural network, this spatial alignment of the sources can be learned from time differences in the signal. This is similar to a delay-and-sum beamforming effect, except the sum is replaced with a deep network.

CB-UNet The output of our lightweight CB-Conv-TasNet often contains audible artifacts (i.e., crackling, static) that reduce the listening experience. Interestingly, these artifacts have little effect on traditional metrics, like Signal-to-Distortion Ratio (SDR), but have a noticeable effect on subjective listening scores (see 5.4.2). These artifacts are often visible in a frequency representation of the audio. Fig. 5.5 shows how CB-Conv-TasNet alone contains noticeable artifacts when compared to the ground truth. To address this, we cascade a lightweight causal UNet [132] which operates on the mel-scale spectrogram of the input audio. This network, which we call CB-UNet, produces a binary mask which is applied to the output of CB-Conv-TasNet. The combined output, shown in Fig. 5.5 as CB-Net, reduces these artifacts.

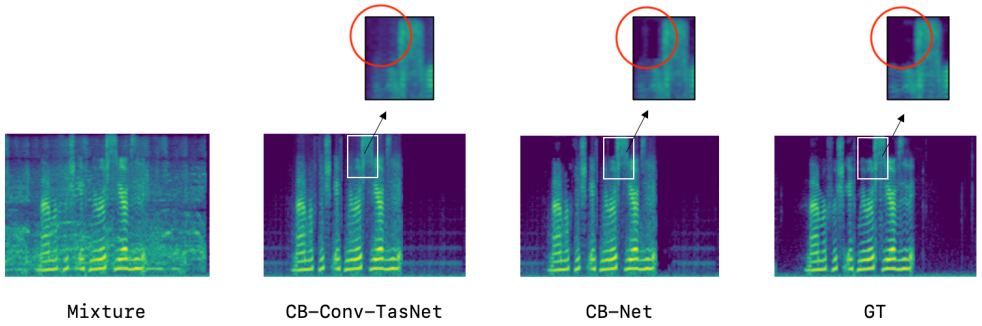


Figure 5.5: *The spectrograms above show the motivation behind a combined time and frequency domain method. The output of the time-domain component, CB-Conv-TasNet, contains artifacts, particularly at high frequencies. Although subtle, these artifacts are perceptible by human listeners. CB-Net is able to reduce these artifacts by using a frequency-domain network (CB-UNet) that masks unwanted frequencies.*

The mean opinion scores in our evaluation shows the strength of the cascaded CB-Net when compared to the time-domain component only.

5.2.3 Neural Network Detailed Description

CB-Conv-TasNet The input to the network is a binaural mixture given by $\mathbf{x} \in \mathbb{R}^{2\times T}$. The first step is an encoder that transforms the mixture $\mathbf{x}$ into $\mathbb{R}^{N\times T/L}$ with a 1D convolution of size L and stride L. This is followed by a ReLU layer. The encoder's outputs are next fed into a temporal convolution network that consists of stacks of 1-D convolutions with increasing dilation factors. We use 14 convolution layers with dilation factors of 1,2,4,..,64 repeated twice, with a ReLU nonlinearity and skip connection after each convolution. The encoder output is multiplied with the output of the temporal conv-net, before being fed through a fully connected Decoder layer which transforms the output back into $\mathbb{R}^{2\times W}$.

In a real-world implementation, we do not have access to the full waveform, but only packets of data at a time. Furthermore, we must process these packets with limited access to future input samples. Given 15.625 kHz sampling rate, we choose to process packets

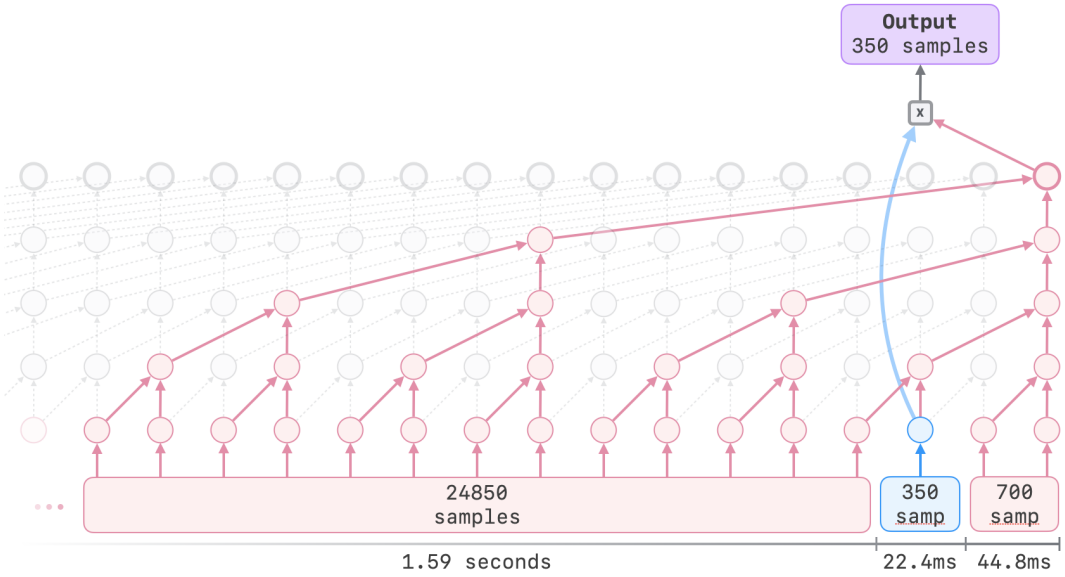


Figure 5.6: *CB-Conv-TasNet, the time-domain component of CB-Net. Given a packet of 350 samples (22.4ms) highlighted in blue, we use 1.5s of past input and 44.8ms of future input to output the separation results. Our caching scheme works as follows: When we receive a new 350ms samples, all intermediate activations (circles in the diagram) slide to the left, and we compute only the rightmost column of outputs.*

of 350 samples at a time (22.4ms), which is our window size W. We also use $2W$, or 700 samples of lookahead time (44.8ms) and 1.5s of past samples. Since we have no padding in the temporal convolution net, the network starts with this large temporal context and outputs exactly $\mathbb{R}^{1\times W}$ samples, corresponding to the desired output for our input packet of W samples. When we receive the next packet of size W, all intermediate activation from the encoder and temporal conv-net can be shifted over by W/L samples and re-used. We chose $L = 50$, but any divisor of W would work. Re-using intermediate outputs from previous packets saves over 90% of the compute time for a new packet in our network.

CB-UNet The frequency domain network is a mono-channel network that outputs a binary mask for each time-frequency bin. The input $\mathbf{x} \in \mathbb{R}^{1\times T}$ is a summation of the binaural left and right channel, which is the equivalent of a broadside beamformer. We first run a STFT, which

is a mel-scale fourier transform with hop size of 350, a window size of 1024 including zero padding on the edges, and a 128 bin mel-scale output. The network input is a spectrogram of 64 time bins and 128 frequency bins, corresponding to a receptive field of 22400 samples, or $1.43s$. In order to maintain the causality requirement, we use the same lookahead strategy as the time-domain network where we allow 700 samples of lookahead for a target packet of 350 samples. The UNet architecture contains 4 downsampling and upsampling layers, starting with 64 channels and doubling the number of channels at each subsequent layer. The downsampling layers contain a depthwise separable convolution followed by a 2×2 max pooling, and the upsampling layers contain a depthwise separable convolution followed by a transposed convolution for upsampling. The output is a sigmoid function, which is then thresholded to return a binary mask in $[0, 1]^{128\times64}$. When outputting a spectrogram mask on an $\mathbb{R}^{128\times64}$ input, we predict a mask over the entire input even though we only need the output for a specific slice of 350 samples, or a $\mathbb{R}^{128\times1}$ mask. Further optimizations could be made by caching intermediate outputs or only computing the mask for the target samples. However CB-UNet's run-time was so small compared to the rest of the network that these optimizations were not necessary.

Combining the Outputs At each time step, the output of CB-Conv-Tasnet is an audio waveform in $x \in \mathbb{R}^{1\times350}$, and the output of CB-UNet is a spectrogram mask in $\mathbf{M} \in \mathbb{R}^{128\times64}$. We run the same fourier transform on the buffered conv-tasnet outputs to produce a spectrogram $\mathbf{X} \in \mathbb{R}^{128\times64}$. Our output can then be computed by $iSTFT(\mathbf{M} \otimes \mathbf{X})$. Our empirical results show that this gives the best results compared to other methods such as ratio masking.

Training CB-Conv-TasNet is trained with an $L1$-based loss over the waveform along with the multi-resolution spectrogram loss. Formally, provided s_0 is our target speaker and x' is the output from the network, our loss is:

$$L(s_0, x') = \|s_0 - x'\|_1 + L_{sc}(s_0, x') + L_{mag}(s_0, x')$$

$$L_{sc}(s_0, x') = \frac{\|STFT(s_0) - STFT(x')\|_F}{\|STFT(s_0)\|_F}$$

$$L_{mag}(s_0, x') = \|\log(STFT(s_0) - \log(STFT(x'))\|_1$$

STFT denotes the magnitude of the short time Fourier transform, and F denotes the Frobenius norm. L_{sc} and L_{mag} represent spectral convergence and magnitude losses, which gave better results than L1 loss alone.

For training CB-UNet, for each time frequency bin, the training target $\mathbf{M}$ is 1 if the target voice is the dominant component, and 0 otherwise. Formally, $\mathbf{M}(f,t) = [\mathbf{S}_0(f,t) \geq \mathbf{S}_i(f,t)], \quad \forall i = (1..n)$. The network is then trained with the binary cross entropy of the output compared to the target mask.

Hyperparameters and Training Details We use a learning rate of 3×10^{-4} along with the ADAM optimizer [78] for training the network. The network was trained on a single Nvidia TITAN Xp GPU. Because of the small size of the network, training could be completed within a single day and generally required ≈50 epochs to reach convergence.As an additional data augmentation step we make the following perturbations to the data: High-shelf and low-shelf gain of up to $2dB$ are randomly added using the `sox` library.

5.2.4 Synchronized wireless earbuds

We seek to capture speech from the target speaker's mouth which sits on the sagittal plane roughly equidistant to the ears. Given an ear-to-ear spacing of 17.5cm, to effectively isolate this central plane we require a distance precision on the order of a few centimeters. An interaural time difference of 100μs would correspond to source maximally 3.43 cm off this central plane, therefore we target a synchronization accuracy under 100μs.

Hardware Our custom hardware design contains a pulse-density modulated (PDM) microphone (Invensense ICS-41350) and a Bluetooth Low Energy (BLE) microcontroller (Nordic

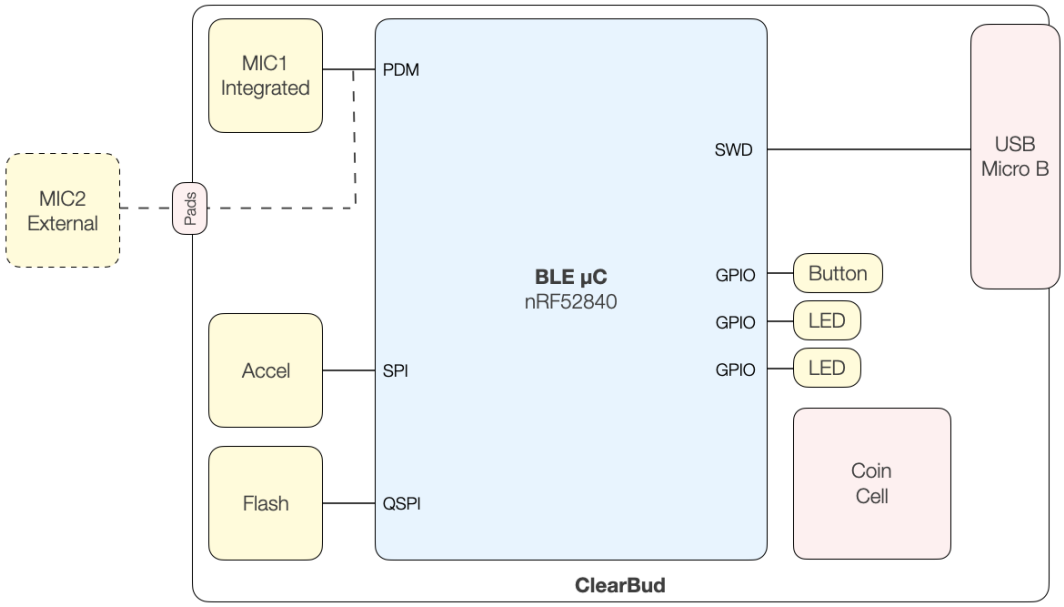


Figure 5.7: *Hardware Block Diagram. Each ClearBud integrates a PDM microphone, accelerometer, flash, and coin cell battery. Buttons and LEDs are used for interfacing with the device, and a USB port is used for programming and debug.*

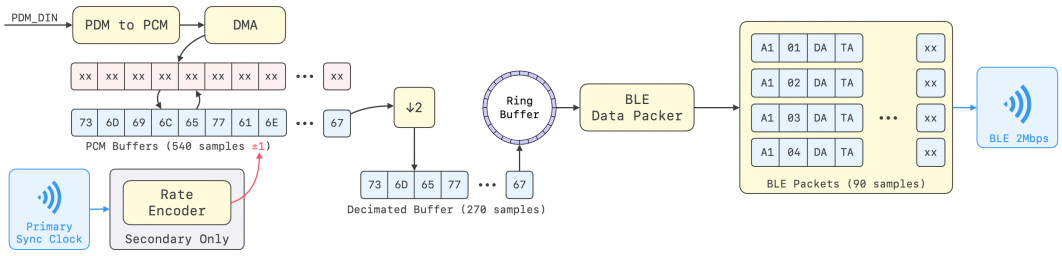


Figure 5.8: *Time Sync Design. The primary ClearBud broadcasts a sync clock over the air. The secondary ClearBud then uses the sync clock to rate encode, by increasing or decreasing the size of its local PCM buffer.*

nRF52840).[1] The system is powered off of a CR2032 coin cell battery and programmed via SWD over a Micro-USB connector. Each ClearBud has an integrated PDM microphone set to a clock frequency of 2MHz. With an internal PDM decimation ratio of 64, this provides us a sampling frequency of 31.25kHz. As most HD voice applications and wideband codecs are limited to 16kHz [34], we decimate further in firmware by a factor of 2, giving us a final sampling frequency of 15.625kHz.

Two 16-bit 180 sample size Pulse-Code Modulation (PCM) buffers are round-robined: one is filled with incoming PCM data while the other is processed. The DMA is responsible for both clocking in the PDM data and converting it into PCM. One buffer is always connected to the DMA, while the other is freed for processing for the rest of the data pipeline. When the buffer connected to the DMA fills, the buffers switch roles and we begin processing data on the newly freed buffer, and connect the other buffer back to the DMA. With this design we always have a continuous PCM stream to operate on. Both ClearBuds transmit the PCM microphone data to a mobile phone for input into our neural network. To maximize throughput, we use the highest Bluetooth rate and packet sizes supported by iOS, which is 2Mbps and 182 bytes, respectively. We design a lightweight wireless protocol where the first 2 bytes represent a monotonically-increasing sequence number, while the other 180 bytes are reserved for the 16-bit PCM audio samples. The sequence number is used on the phone so that we can zero-pad PCM data in the occasional event that a packet is dropped either over-the-air or by the radio hardware. This zero-padding keeps the left and right microphone data aligned on the host side in areas of poor radio performance or interference in the environment.

The hardware schematic and layout for ClearBuds was designed using the open source eCAD tool KiCad. A 2-layer flexible printed circuit was fabricated and assembled by PCBWay. The 3D printed enclosures were designed using AutoDesk Fusion 360 and printed with a Phrozen Sonic Mini using a liquid resin fabrication process. The MEMS microphone sits

[1]For future research applications, an ultra low-power accelerometer (Bosch BMA400), a 1Gbit NAND flash for local data collection (Winbond W25N01GVZEIG), and support for speaker and an additional microphone are included.

behind the lid on the earbud's outer surface. A single button on the enclosure provides access to turn on and off the earbuds.

Microphone synchronization Three components are necessary for maintaining microphone synchronization: (1) As each of our earbuds has its own local clock source, we need to establish a common clock between them so that they have the same reference of time, (2) a synchronized startup so each earbud starts recording from their respective microphone at the exact same time, and (3) a rate encoding scheme to control the earbud's sampling rate to match each other.

In our system, each earbud has its own respective 32MHz clock source with a total +/- 20ppm frequency tolerance budget. So, in the worst case scenario, the earbuds will have 2.4 milliseconds of drift each minute. We use the Nordic's TimeSlot API [22], which grants us access to the underlying radio hardware in between Bluetooth transmissions. This provides us a transport to transmit and receive accurate time sync beacons [11]. Each ClearBud keeps a free-running 16MHz hardware timer with a max value of 800,000, overflowing and wrapping around at a rate of about 20 Hz. One ClearBud is assigned as the timing host, while the other ClearBud will synchronize its free-running timer to the host's. The primary ClearBud (timing host) transmits time sync packets at a rate of 200 Hz. These packets contain the value of the free-running timer at the time of the radio packet transmission. When the secondary ClearBud receives this packet, it can then add or subtract an offset to its own free-running timer for a common clock.

Once each ClearBud is connected to the mobile phone, the phone sends a `START` command to both ClearBuds over BLE. Each ClearBud contains firmware which arms a programmable peripheral interconnect (PPI) to launch the PDM bus once the 16MHz free-running timer wraps around at 800,000. By using this method, we bypass the CPU and trigger a synchronized startup entirely at the hardware layer. One caveat is that the mobile phone could write to one ClearBud right *before* its clock wraps around at 800,000, and the other ClearBud right *after* it wraps around at 800,000. With a clock that wraps around at 20Hz, this would

trigger a mismatched startup and cause an alignment error of 50ms. To correct for this, each ClearBud reports its common clock timer value to the phone once it has received the `START` command. The phone can then remove the first 781 audio samples (781 samples / 15.625kHz = 50ms) if one ClearBud started streaming 50ms before the other.

The final component to keeping the audio streams aligned is to create a rate encoding scheme between the ClearBuds. With the time sync beacons from the primary ClearBud, the other ClearBud now has both its local clock and the common clock (primary ClearBud's local clock). With these two clocks, the secondary ClearBud can identify how much faster or slower its PDM clock is running in relation to the primary ClearBud. We note that with a 2MHz PDM clock and a PDM decimation ratio of 64, each audio sample occupies 32 us. The non-primary ClearBud can then add or remove a sample to its PDM buffer every time the difference between the clocks exceeds a multiple of 32 us. By doing this, the secondary ClearBud ensures that its PDM buffer starts filling up at the exact same time as the primary ClearBud's PDM buffer, with a tolerance of 32 us.

5.3 Training methodology

Training the network in a supervised way requires clean ground truth speech samples as training targets. This is difficult to obtain in fully natural settings since the ground truth speech is corrupted with background noise and voices. Training a network that generalizes to in-the-wild scenarios also requires the training data to mimic the dynamics of real speech as closely as possible. This includes reverb, voice resonance, and microphone response. Synthetically rendered spatial data is the easiest type of data to obtain, but most different from real recordings, while real speakers wearing the headset in an anechoic chamber provide the best ground-truth training targets, but are the most costly to obtain. Synthetic data can simulate various reverb and multipath that are not captured in an anechoic chamber. We adopt a hybrid training methodology where we first train on a large amount of synthetic data and fine-tune on real data recorded with our hardware. Our training method is based on the commonly used mix-and-separate framework [189], where clean speech and noise samples

are recorded separately and combined randomly to form noisy mixtures. Our results show that our network trained this way generalizes to naturally recorded noisy data in real-world environments.

Synthetic data. This type of data is the easiest to obtain, since a wide variety of voice types and physical setups can be generated instantly. Many machine learning baselines, e.g., [95, 66, 155], only train and evaluate on synthetic data generated in this manner. To generate the synthetic dataset, we create multi-speaker recordings in simulated environments with reverb and background noises. All voices come from the VCTK dataset [160] (110 unique speakers with over 44 hours), and background sounds come from the WHAM! dataset [172], with 58 hours of recordings from a variety of noise environments such as a restaurant, crowd, and music.

To synthesize a single example, we create a 3-second mixture as follows: two virtual microphones are placed 17.5 cm apart, which is the average distance between human ears [130]. The target speaker's voice is placed at the center between the two virtual microphones, and a second voice is placed randomly between 1 and 5 meters away and at a random angle. A randomly chosen background noise is also placed in the scene. We then simulate room impulse responses (RIRs) for a randomly sized room using the image source method implemented in the pyroomacoustics library [2, 137]. The room is rectangular with sides randomly chosen between 5 and 20 meters, and the RT60 values are randomly chosen between 0 and 1 second. All signals are convolved with the RIR and rendered to the two channel microphone array. The volumes of the background are randomly chosen so that the input signal-to-distortion ratio is roughly between -5 and 5 dB. For training, we use 10,000 mixtures generated in this manner.

Hardware data. While a large amount of synthetic data can be easily rendered to train the network, it does not contain characteristics such as the microphone response of physical hardware and imperfections in the time-of-arrival. To address this, we also train on a set of recorded voice samples from our earbuds. We set up a foam mannequin head with an artificial mouth speaker (Sony SBS-XB12) that plays VCTK samples as the spoken ground

truth. For background voice recordings, the speaker is placed in varying locations within a one meter radius of the foam head. Physically recorded background noise is provided by binaural version of the WHAM! dataset [172], which was recorded in real environments using a binaural mannequin like ours. We record 2 hours each of clean speech, and background voices. 2000 random mixtures are then created for training.

Human data. The spoken hardware data above still does not contain natural voice resonance since it is played out of an electronic speaker. Furthermore, the background sounds recorded by a mannequin wearing earbuds still misses some of the physical filtering of the human body. To better capture desired output of real scenarios, we collect a ground-truth speech dataset in an anechoic chamber with human speakers (5 male, 4 female) and a noise dataset in real environments with human listeners. For the voice data, each human speaker wore our ClearBuds prototypes, and uttered 15 minutes of text from Project Gutenberg in the anechoic chamber. The purpose of this anechoic data is to provide clean training targets for the network, modelling the resonance of human speakers wearing our hardware. For the real world noise dataset, individuals wore ClearBuds and recorded various noisy scenarios such as washing dishes, loud indoor/outdoor restaurants, and busy traffic intersections. 2000 random mixtures of clean voice and recorded noise were generated for this dataset.

Our network is jointly trained using all these datasets. Note that testing and evaluation is done *outside* the anechoic chamber.

5.4 Experiments and Results

We first compare our end-to-end system performance against a commercial wireless earbud system. We then present in-the-wild evaluation of our system. Next, we compare numerical results against various speech enhancement baselines. Finally, we present system-level evaluations. Our work is approved by the IRB.

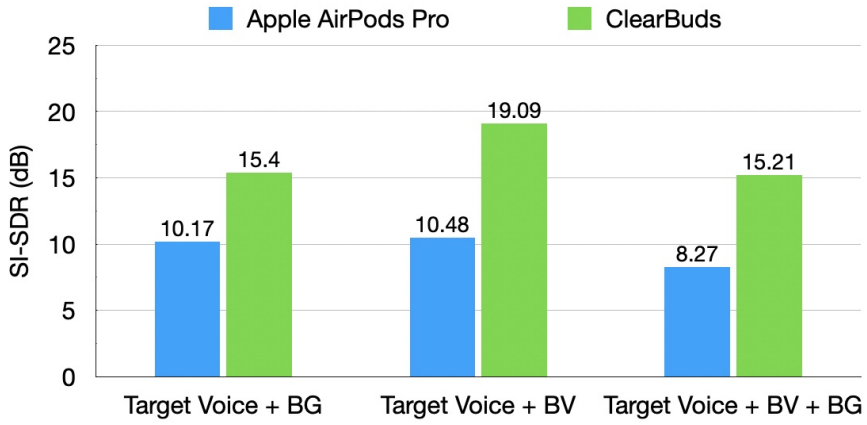


Figure 5.9: *Comparison with AirPods Pro. Reporting the output SI-SDR (note: not SI-SDR increase). ClearBuds exceeds in three conditions: target voice plus background noise (BG), target voice plus background voice (BV), target voice plus background voice and noise.*

5.4.1 Comparison with Beamforming Earbuds

We evaluate our end-to-end system against the Apple AirPods Pro headset connected to a iPhone 12 Pro in a repeatable physical set up. In our evaluation, as is typical, there is no overlap between training and test datasets.

Procedure. We use the popular metric *scale-invariant signal-to-distortion ratio* (SI-SDR) [134]. While SI-SDR provides a repeatable metric used in the acoustic community, it requires a clean, sample-aligned ground truth (target voice) as the basis for evaluation. Therefore, we create a repeatable soundscape for our test setup where a sample-aligned ground truth can be obtained. A foam mannequin head with a speaker (Sony SBS-XB12) inserted into its artificial mouth uttered one hundred VCTK samples with identities and samples unseen in the training set. The mannequin wore ClearBuds and AirPods Pro in subsequent experiments, and the outputs of the two systems could be directly compared. Ambient environmental sound (from WHAM! dataset) was played via four monitors (PreSonus Eris E3.5) positioned to fill 3 meter by 4 meter room, and background voice (also VCTK) was

(a) (b) (c) (d)

Figure 5.10: *In-the-wild experiments in various scenarios (crowded cafe, busy intersection, outdoor plaza, classroom) were conducted across 8 users and indoor and outdoor environments, all unseen in our training dataset.*

played from a monitor positioned 0.4 meters from head on the right. All speakers were driven through a common USB interface (PreSonus 1810c) ensuring the same time-alignment and loudness between the two test conditions. Since Apple AirPods Pro beamforming cannot be toggled on and off, we cannot calculate an SI-SDR increase (SI-SDRi), and therefore report output SI-SDR. To establish the ground truth voice against which to calculate SI-SDR, we record clean target voice through each headset. Ambient noise SNR ranged between 0dB and 16dB with respect to target voice. Qualitatively, this sounded like a second person speaking loudly in a noisy bar or cafe. Finally, background voice SNR ranged between 6dB and 12dB, qualitatively sounding like a person speaking from a meter or two away.

Results. We report output SI-SDR from the two systems in Fig. 5.9. To calculate output SI-SDR, we align individual one second chunks and take the logarithmic mean across 250 chunks. We find that ClearBuds achieves higher output SI-SDR across all test conditions when compared to the beamforming utilized by the Apple AirPods Pro. For a qualitative comparison of AirPods Pro versus ClearBuds performance with human speakers, see video: https://clearbuds.cs.washington.edu/videos/airpods_comparison.mp4.

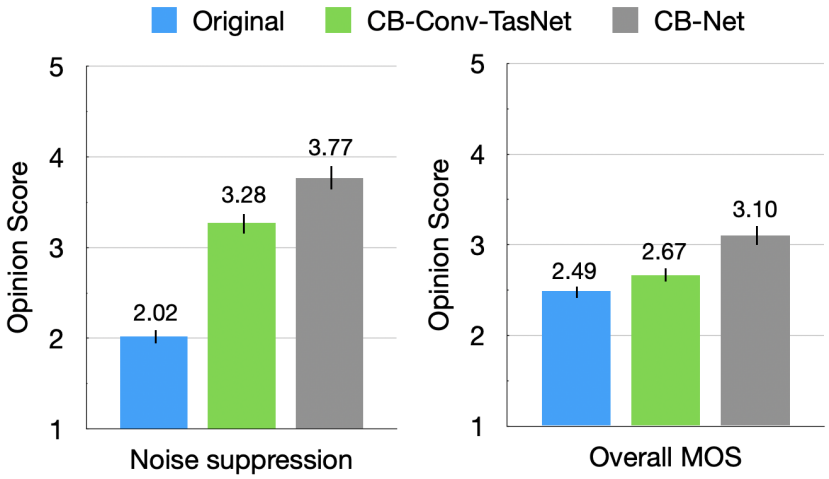


Figure 5.11: *In-the-wild study results. Noise suppression indicates perceived quality of background noise reduction (higher is less intrusive). Overall MOS indicates overall perceived quality. Error bars are 95% CI.*

5.4.2 In-the-Wild Evaluation

We perform in-the-wild evaluation in indoor and outdoor scenarios as well as users not in the training data. The procedure and results are described in the following sections.

In-the-wild experiments. Eight individuals (four male, four female, mean age 25) with a variety of accents wore a pair of ClearBuds and read excerpts from Project Gutenberg [123] while in four noisy environments: a coffee shop, a noisy intersection, an outdoor plaza, and a classroom (see Fig. 5.10). The environments featured ringing phones, cross-talk from other people, ambient music, a crying baby, opening/closing doors, driving vehicles, and street noise, amongst other common sounds. These experiments were uncontrolled in that the background voices and noise were naturally occurring sounds that are typical to these real-world scenarios and were mobile.

Evaluation procedure. In-the-wild evaluation precludes access to clean, sample-aligned truth to compute SI-SDR. Instead, the common (and expensive) procedure is to perform a user study and compute the mean opinion score. Since this is a time-consuming process, prior works on binaural networks, e.g., [95, 152, 66], avoid in-the-wild evaluation. Since our goal is

(a) (b)

Figure 5.12: *Mobility of speaker and noise sources in-the-wild. Red box highlights a moving truck on the road while ClearBuds user is walking. Video: https://youtu.be/HYu0ybjcQPA?t=127*

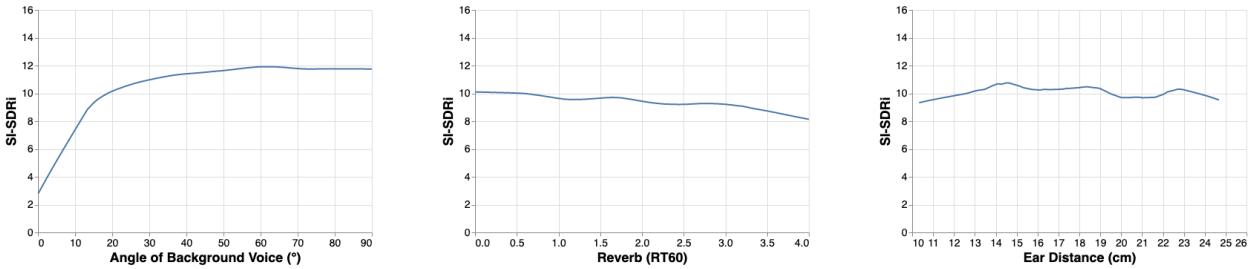
to design and evaluate an in-ear system in real scenarios, we recruit thirty-seven participants (11 female, 26 male, mean age 29) for a user study. Each participant listened to between 6 and 11 in-the-wild audio samples (avg. 9.38 samples, each between 10–60 seconds). Each speech sample was processed and presented three ways: (1) the original input, (2) CB-Conv-TasNet, and (3) CB-Net, yielding a total of $37 \times 9.38 \times 3 = 1{,}041$ rating samples.

Participants were encouraged to use audio equipment they would typically use for a call. Fourteen used earbuds, thirteen used computer speakers, seven used headphones, and three used phone speakers. The study took about 25 minutes per participant. As is typical with noise suppression systems, participants were asked to give ratings in two categories: the intrusiveness of the noise and overall quality (mean opinion score - MOS):

1. **Noise suppression:** *How INTRUSIVE/NOTICEABLE were the BACKGROUND sounds? 1 - Very intrusive, 2 - Somewhat intrusive, 3 - Noticeable, but not intrusive, 4 - Slightly noticeable, 5 - Not noticeable*

2. **Overall MOS:** *If this were a phone call with another person, How was your OVERALL experience? 1 - Bad, 2 - Poor, 3 - Fair, 4 - Good, 5 - Excellent*

Results. Fig. 5.11 shows the noise intrusiveness and MOS values for the original



(a) (b) (c)

Figure 5.13: *(a) Performance against angle of background voice in presence of significant multipath. (b) Performance against amount of reverberation in an indoor room. RT60 (in seconds) measures how long sound takes to decay by 60 dB in a space with a diffuse soundfield. (c) Performance as distance between ears increases.*

microphone, CB-Conv-TasNet, and CB-Net. As expected, applying CB-Conv-TasNet to the original audio helped suppress noise dramatically, increasing opinion score from 2.02 (slight better than *2 - Somewhat intrusive*) to 3.28 (between *3 - Noticeable, but not intrusive* and *4 - Slightly noticeable*) ($p<0.01$). The light-touch, spectrogram-masking clean up method featured in CB-Net increased noise suppression opinion score significantly ($p<0.001$) to 3.77, indicating the method did indeed further suppress perceptually annoying noise artifacts. Importantly, this step also increased overall MOS. While users only slightly preferred ($p<0.05$) CB-Conv-TasNet (2.67) to the original input (2.49) due to artifacts introduced, they more significantly ($p<0.001$) preferred our CB-Net (3.10), an increase of 0.61 opinion score points from the input. For context, in the flagship ICASSP 2021 Deep Suppression Noise Challenge [126], with state-of-the-art, real-time algorithms run on a quad-core desktop CPU, the winning submission increased MOS by 0.57 [105] from input.

Note that in our in-the-wild experiments, the background noise and voices were not static. The speakers themselves can also be mobile (see Fig. 5.12). Our network was able to adaptively remove the background noise and achieve speech enhancement with mobility.

Table 5.1: *Benchmarking our neural network. We show results for a target voice speaking in three noise scenarios: (1) Background noise (BG), (2) Background voice (BV), and (3) Background noise and background voice (BG and BV). CB-Conv-TasNet performs slightly better on synthetic data, but as shown in Fig. 5.11, does not generalize as well to in-the-wild scenarios. This demonstrates the importance of evaluating networks on real in-the-wild hardware data.*

	SI-SDR increase (SI-SDRi)			Output PESQ		
Method	Target with BG	Target with BV	Target with BV + BG	Target with BG	Target with BV	Target with BV + BG
CB-Net	10.41	10.56	9.35	2.08	2.68	1.81
CB-Conv-TasNet	11.19	11.01	9.68	2.24	2.58	1.91
CB-Conv-TasNet Single Mic	6.15	0.13	2.34	1.82	1.84	1.53
CB-UNet	3.21	0.78	1.82	1.60	2.10	1.50
DTLN [170]	7.02	0.06	2.13	2.08	1.95	1.67
Causal Demucs [35]	6.62	-0.03	2.11	1.80	1.88	1.43
Ideal Ratio Mask (IRM, oracle)	11.41	11.53	12.04	2.53	3.00	2.44
Ideal Binary Mask (IBM, oracle)	9.97	11.05	10.85	2.30	2.90	2.21

5.4.3 Benchmarking our Neural Network

The conventional evaluation in the machine learning and acoustic community is to evaluate models and techniques on synthetic data against baselines. For completeness, we compare our method against a variety of speech enhancement baselines using the synthetic dataset. For evaluation, an additional 1000 mixtures of 3 seconds each were generated such that there was no overlapping identities or samples between the train and test splits.

Evaluation Procedure. For comparisons to other baseline methods, we use the popular SI-SDR and PESQ metrics. Unlike the AirPods experiment, where the original noisy mixture could not be recorded since AirPods beamforming cannot be toggled off, here we compute SI-SDR of the ground truth relative to both the input noisy mixture and then to the network output. When reporting the increase from the input SI-SDR to output SI-SDR, we use the SI-SDR improvement (SI-SDRi).

For a deep learning baseline in the waveform domain, we choose the causal Demucs model [35]. This is a single channel method which was recently shown to outperform many other deep learning baselines and runs real-time on a laptop CPU. We also compare with Dual-signal Transformation LSTM Network (DTLN) [170]. This method also runs on a laptop or mobile phone in real-time. To compare with spectrogram based methods, we use the oracle baselines, ideal ratio mask (IRM) and ideal binary mask (IBM) [147, 164], that use the ground truth voice to calculate the best possible result that can be obtained by masking a noisy spectrogram.

As an ablation study, we report results with each individual component of the network, *CB-Conv-TasNet* and *CB-UNet*. We also show results when the multi-channel part of our network, *CB-Conv-TasNet*, only has access to one microphone, labeled as *CB-Conv-TasNet Single Mic*. This explicitly shows the advantage of using two microphones. There are only a few deep learning methods that tackle binaural speech separation for mobile processing, and the most relevant ones, such as [152] and [55], do not have publicly available code to test against.

Results. As shown in Table 5.1, our binaural method is comparable to the best possible results that can be obtained by a spectrogram masking method (IBM, IRM). We also show an improvement over waveform based deep learning methods that only use a single microphone input. In particular, the improvement is greatest when there are two speakers present (Target Voice + Background Voice). This is because single channel methods can only rely on voice characteristics, whereas our network also uses spatial cues to separate the speaker of interest. Although *CB-Net* shows similar or worse performance to *CB-Conv-TasNet*, subjective evaluation on in-the-wild hardware data shows that *CB-Net* is far superior to human listeners (see 5.4.2).

Examples of the synthetic dataset, outputs from all the methods and qualitative comparisons against Krisp [84], a commercial noise suppression system, can be found linked from our project website: https://clearbuds.cs.washington.edu.

Additional neural network evaluations We numerically evaluate various aspects of the design by changing the angle of background voice, reverberance in the environment, and microphone separation.

Angle of background voice. The ability of our network to separate the target voice from a background voice is based on utilizing the time difference of arrival to the binaural microphones. Because we only have two microphones, this ability is limited when the background voice is in the front-back plane of the speaker. In this case, the background voice will arrive at each microphone simultaneously, and there will be no spatial cues to separate the two voices. To illustrate this effect, we graph the separation performance as a function of the angle of the background voice in Fig. 5.13(a).

Multipath and reverberant environments. While our in-the-wild experiments show the performance in various indoor and outdoor environments, we benchmark our system in different reverberant conditions, including those more reverberant than seen during training. Synthetically generated mixtures are generated using the pyroomacoustics library with the RT60 value randomly chosen between 0 and 4s. We generate 200 examples and plot the

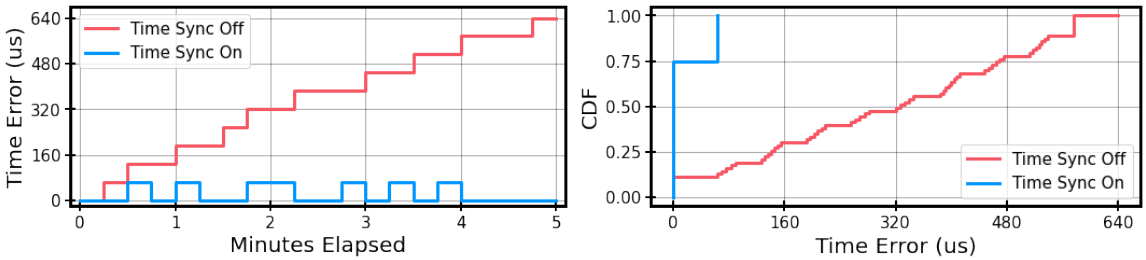


Figure 5.14: *Time Synchronization Validation. Without time synchronization (red), microphone samples drift apart and lose alignment at about 128µs/min.*

SI-SDRi compared to the RT60 in Fig. 5.13(b). Our method shows only a slight decrease in performance as the reverberation of the environment increases. Because the target speaker is physically close to the microphone array, our setup is generally less affected by reverberations than other kinds of source separation problems where the target speaker may be further away.

Separation between microphones. Our in-the-wild evaluation across 8 participants showed generalization across facial features. Here, we benchmark our method to different head sizes where the distance between the microphones may be different. We generate 200 synthetic samples, where the distance between the microphones is randomly chosen between 10 and 25 cm. Because the target speaker is in the middle of the microphone array, the target signal will arrive at both mics simultaneously, regardless of the microphone distance. Fig. 5.13(c) show little change in performance even with microphone distances greatly different than used during training.

5.4.4 System Evaluation

Synchronization. In order to evaluate this, we place both ClearBuds roughly equidistant from a speaker. A click tone is played every 15 seconds for 5 minutes, and recorded on both ClearBuds with time sync disabled and enabled. We calculate the sample error on each recorded click offline and convert it into time error with a sampling rate of 15.625kHz. Fig. 5.14(a) shows the synchronization results across a five-minute interval. With time sync enabled, the sample error never exceeds 1 sample at 15,625 kHz, or 64 μs. Fig. 5.14(b) also

Table 5.2: *Neural network run time on smartphones*

Device	Conv-TasNet	CB-Conv-TasNet	**CB-Net**
iPhone 12 Pro	155.5ms	17.5ms	21.4ms
iPhone 11	165.4ms	18.6ms	22.7ms
iPhone XS	241.5ms	27.2ms	33.0ms
FLOPs/packet	1078M	97M	131M

shows the CDF of the timing error across experiments of 5 minutes each conducted with other Bluetooth devices in the environment, with and without time synchronization.

Run-time and end-to-end latency. Mouth-to-ear delay is defined as the time it takes from speech to exit the speaker's mouth and reach the listener's ear on the other end of the call. The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) G.114 recommendation regarding mouth-to-ear delay indicates that most users are "very satisfied" as long as the latency does not exceed 200 ms [62]. In our end-to-end system, we targeted a one-way latency of 100 ms prior to uplink, leaving up to 100 ms of network delay to move an IP packet from the source to the destination.

With a 180-sample PCM buffer being filled at 31.25 kHz, there is a 5.76 ms delay prior to the samples reaching BLE stack. Once these samples reach the radio hardware, there is a worst-case additional latency of 7.5 ms as defined by the minimum BLE connection interval supported by Bluetooth 5.0 [16]. At the time of writing, the latest iOS supports a minimum BLE connection interval of 15 ms. After the samples reach the mobile phone, we wait for 67.2 ms to receive enough samples to run a forward pass of our network. Our network has a run-time of 21.4 ms on an iPhone 12 Pro (see Table 5.2). The number of FLOPs is computed over each packet of 350 samples. Together, we have a latency of 109 ms, leaving 91 ms for one-way network delay (RTT=182ms).

Power analysis. CB-Net uses an order of magnitude lower FLOPs per second compared to

Table 5.3: *ClearBuds hardware power consumption*

Component	Power Consumption
BLE SoC (nRF52840)	12.02 mW
Microphone (ICS-41350)	0.77 mW
Ideal Diode (LM66100DCKT)	0.27 μW
Buck Efficiency Loss (MAX38640)	1.75 mW
Total	14.54 mW

Conv-TasNet on the smartphone, significantly reducing the computational and corresponding power consumption. We also measure the power consumption of the ClearBuds hardware. We measure current consumption by powering our system through its Micro-USB port with a DC power supply set to 3V, which goes through the same power path as our coin cell battery. While continuously wirelessly streaming microphone data, we measure average current consumed to be 5 mA. With the CR2032's nominal capacity of 210 mAh, this translates to approximately 42 hours of operation. Table 5.3 shows a breakdown by component of the system's power consumption. The accelerometer (BMA400) and flash (W25N01GVZEIG) are omitted as they are power gated during streaming.

5.5 Limitations & Future work

The first limitation is that the user must be wearing both wireless earbuds to benefit from our binaural noise suppression network. Second, with only two microphones, there is an opportunity for background voices to remain in the uplink channel if the voice is within a few degrees of the target speaker's sagittal plane (see Fig. 5.13(a)). The underlying assumption of our network is that the mouth is in the middle of the user's ears, though as seen in Fig. 5.13(c) and our in-the-wild evaluation, some variance is permissible.

While we minimize the power consumption of the ClearBud hardware, we shift the processing and therefore power consumption to the more powerful mobile phone. Performing

network computation on the mobile phone over a cloud GPU is an enhancement in terms of user privacy and security so that sensitive voice data is not transmitted to the cloud. While mobile chips are becoming more power efficient, an alternative design to explore is to run our neural network on a plugged-in edge device (e.g., router), minimizing computation while achieving similar latency.

Future work could integrate two microphones in each earbud, so that each earbud could beamform toward the user's mouth prior to processing in the neural network. We also had to develop a custom wireless audio protocol to stream two microphones to a single phone. While this prevents this architecture from being deployed on today's commodity wireless earbuds, adoption may be imminent as Bluetooth 5.2 shows promise with the introduction of Multi-Stream Audio and Audio Broadcast [15].

Our network could also be deployed on other multi-microphone mobile or resource-constrained edge systems such as smartwatches, augmented reality glasses, or smart speakers to allow for enhanced voice control or telephony in noisy environments. The hardware and firmware for Clearbuds could be leverage to produce wireless, synchronized microphone arrays for telephony, acoustic activity recognition or for swarm robot localization and control.

5.6 Conclusion

Real-time speech enhancement has been an open research challenge for multiple decades. The recent proliferation of wireless earbuds and neural network architectures provides an opportunity to build systems that bridge neural networks and wireless earbuds to create new capabilities. Here, we present ClearBuds, the first deep learning based system to achieve real-time speech enhancement with binaural wireless earbuds. At its core is a new open-source wireless earbud design capable of operating as a synchronized binaural microphone array and a lightweight cascaded neural network. In-the-wild experiments show that ClearBuds can achieve background noise suppression, background speech removal, and speaker separation using wireless earbuds.

Chapter 6

FLOWRING

This chapter discusses FlowRing, a ring-form-factor device with processing algorithms that enables interactions across a range of ad hoc surfaces. By scaling its interaction capability across always-available surfaces, FlowRing allows for precise control of spatial computing interfaces across user postures, tasks, and scenarios. An illustrative video figure can be accessed here: `https://youtu.be/t225nkxinaU`.

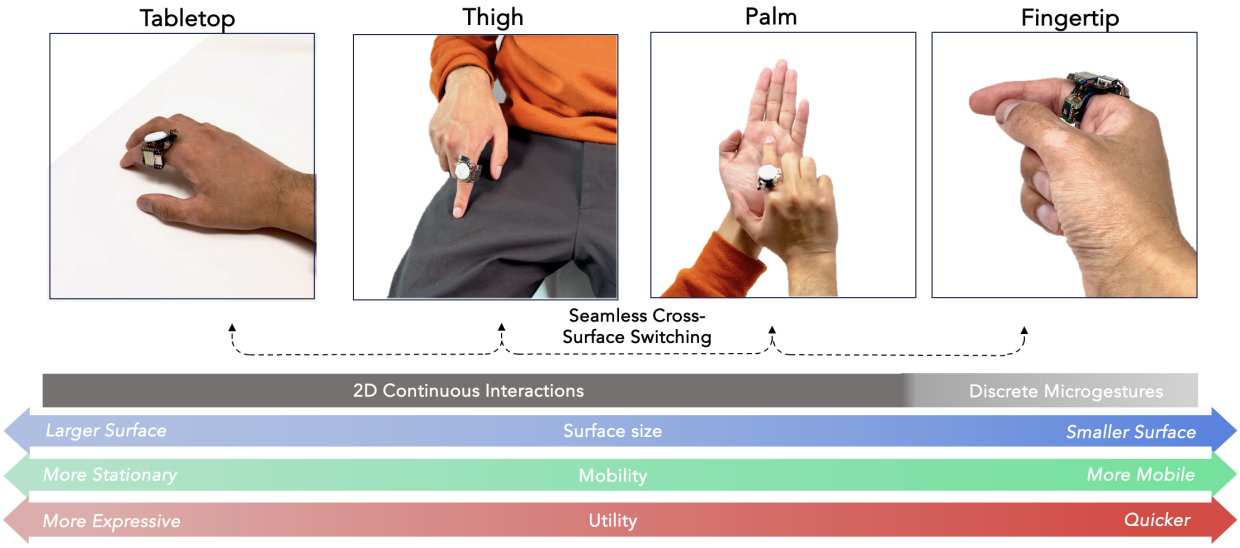


Figure 6.1: *FlowRing enables both precise on-surface interactions on a desktop, pant leg, and palm surface as well as quick, subtle in-air microgestures on the fingertip, facilitating constantly available, ad hoc interactions. This type of peripheral could control mixed reality glasses, where interfaces range from precise desktop experiences to simple, discrete on-the-go widgets.*

6.1 Introduction

Surface-bound interaction offers many desirable characteristics for input: it is ergonomic, precise, subtle, and self-haptic, i.e., uses one's own body for physical feedback. These benefits make trackpads and mice the de facto standards for desktop input today and touchscreens the primary input modality for mobile phones and tablets. However, surface interaction is also inherently constrained, requiring an appropriate surface for the interaction to occur: mice require a desk, and trackpad and touchscreens require digitizing the surface itself, limiting the interaction space to the instrumented screen surface. These constraints pose a challenge for future XR devices in particular, which are evolving for use in both desktop-like experiences (Apple Vision Pro, Meta Quest Pro) as well as on-the-go smartglasses experiences (North Focals, Snapchat AR Spectacles, Oppo Air Glass 2). Users will increasingly move between their desk, their couch, and the broader (on-the-go) environment, impacting the surfaces available to them.

To this end, we propose FlowRing, a novel ring-form factor device that supports seamless on-surface interactions on commonly accessible surfaces: a desktop, pant leg, palm, and fingertip. For larger surfaces, we enable mouse- or touchscreen-like continuous and precise 2D interaction and selection. For the smallest surface, a fingertip, we target mobile usage via microgestures; subtle motions of the index and thumb fingers have been proposed as a compelling way to control on-the-go devices due to their low fatigue rate, high precision, social discretion, and constant availability [10].

Importantly, FlowRing features *seamless* switching between surfaces. Its gating model rejects non-gestural motion and can detect the mode of interactions, smoothly transitioning between different surfaces and microgesture interactions. This capability opens the interaction space for single- and cross-device input. For instance, the user can instantly start 2D interactions on one surface (e.g., a desk) without calibration or gating criteria and then seamlessly move to other available surfaces (e.g., a pant leg) and/or perform microgestures without requiring additional input. Different surfaces or gestures can be mapped to different

UI affordances (e.g., a desktop sketching canvas or a color picker on a palm) or different devices (e.g., cursor control on an XR headset or volume control of earbuds via microgesture swipes on a fingertip). *To our knowledge, FlowRing is the first ring device that enables both continuous on-surface interaction and microgestures as well as the first device to facilitate seamless transitions across these input techniques and across surfaces.*

To power its interactions, FlowRing uses a miniature optical flow sensor, a skin-contact microphone, and an inertial measurement unit (IMU). These sensing modalities are inherently complementary, with the optical flow sensor and IMU capturing finger motion while the contact microphone captures finger contact.

Further, FlowRing's form factor outperforms previous systems in terms of ergonomics. Specifically, unlike other microgesture and surface interaction devices positioned at the fingertip, fingernail, or incorporate multiple rings, FlowRing rests at the finger's base and is completely wireless, resembling conventional jewelry. This design consideration not only ensures comfort but also makes the device socially appropriate for a wider range of settings.

In this chapter we contribute FlowRing, a wireless ring device with optical flow, a contact microphone, and an IMU that enables *both* on-the-go subtle input and in-situ expressive input, and seamless switching between them. We evaluate within an 11-person user study demonstrating the effectiveness of FlowRing's microgesture input capabilities across users with different hand sizes and skin tones as well as a Fitts' law evaluation of continuous 2D surface interaction. Finally, we contextualize our interaction techniques within a set of example scenarios, demonstrating the potential of mobile, seamless cross-surface interaction.

6.2 Interactions

As users change environments and activities, the surfaces available to them may change. In this chapter, we target four commonly available surfaces for FlowRing use: a tabletop, pant leg, palm, and fingertip. These surfaces span different affordances along multiple dimensions, including their size, orientation, and number of hands required for interaction (see Fig. 6.1). A large surface like a desktop would be available while seated and permit expressive, mouse-like

interactions. Similarly, the thigh area of a pant leg may be less precise but offers a similarly large area for continuous cursor control. A medium surface like the hand palm affords touchscreen-like interactions, with one hand performing 2D continuous interactions on the other. Finally, a small surface such as a fingertip can enable subtle one-handed interactions, like discrete microgestures, which can be performed in standing or on-the-go scenarios.

Because future spatial computing devices will be used across scenarios and environments, we similarly target FlowRing to *seamlessly* support input modalities across these dimensions. FlowRing is designed to recognize these surfaces, allowing modality switches to occur without friction. Broadly, FlowRing supports two classes of interactions: 2D continuous surface input and microgestures, which we describe below.

6.2.1 2D Continuous Surface Input for Expressive Interaction

For scenarios featuring a richer user interface and larger surface, our system offers rich input through a trackpad-like 2D continuous input mechanism.Users can conveniently employ a desk while seated, use their clothing for spontaneous input while standing or lounging, or choose their palm for an ad hoc additional control surface. This last method involves using the index finger to contact and slide across the chosen surface.

Importantly, despite being mounted on the finger, our system can enable mouse-like interaction since it supports a tap-to-select gesture with the index finger. This poses a technical challenge to maintain cursor stability during finger lift and attack (see Sec. 6.4.1), and previous 2D interaction systems [72] do not support this capability. By enabling complete 2D continuous interactions across these surfaces, the system can also implicitly support other simpler interactions on surfaces, such as touchscreen-like swipe interactions and discrete gestures.

6.2.2 Microgestures for Quick Interaction

The smallest surface our system supports is the fingertip. Given the size constraints of this interaction space, we target thumb-to-index finger microgestures [10], well suited for

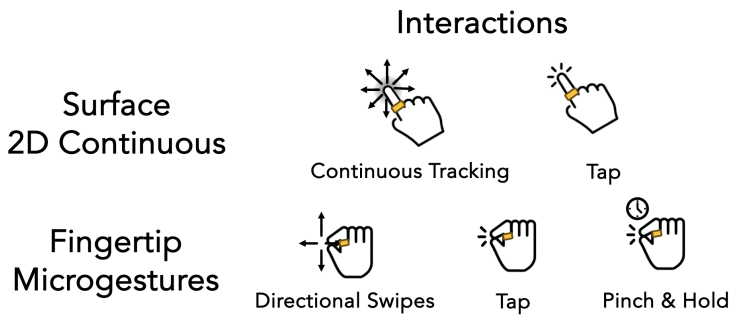


Figure 6.2: *A summary of interactions afforded by FlowRing.*

on-the-go and public use because they are subtle, ergonomic, and quick. The system facilitates five distinct gestures: four cardinal directional swipes, enabling efficient navigation through hierarchical menus, and a tap gesture for making selections. These interactions can efficiently navigate the type of UIs found on smartglasses and other mobile UIs. The gestures are executed using the thumb on the near-tip region of the index finger; these two fingers' natural proximity and agility contribute to the effortless execution of these gestures, enabling users to perform actions with one hand 6.1.

Additionally, the system incorporates a *stateful pinch* feature, defined as the continuous detection of contact between the thumb and index finger. This detection is achieved by monitoring the touch-down and touch-up events of the thumb on the index finger. The stateful pinch adds a temporal aspect to the tap interaction, facilitating more nuanced inputs such as long taps.

6.3 FlowRing Prototype

Our FlowRing design highlights practicality. We chose low-power sensors in low-profile mounting positions. FlowRing consists of three main sensing components: an optical flow sensor, a contact microphone, and a 6-DOF IMU. For the optical flow sensor, we adapted the Pixart PAT9130,[1] a component usually used in industrial applications to monitor shaft

[1]https://www.pixart.com/

rotation. For our form factor, the large depth-of-field working distance was ideal for tracking centimeters away from the surface, and the integrated VCSEL enables a small total package size (4.6 mm by 4.4 mm by 0.3 mm). We mounted the chip on a custom power management and conditioning PCB.

We collect five features from the sensor at approximately 130 Hz: Δx and Δy values, quality of image keypoints, shutter duration, and average brightness of the auto-exposed frame. Together, shutter and brightness provide a proxy for surface z-distance; however, in practice, they are confounded by surface reflectivity, texture, and sensor angle.

We experimented with two alternative mounting positions for the sensor: (1) mounted to the radial side of the proximal phalanx rotated 45 degrees distally, and (2) mounted on the palmar (bottom) side of the proximal phalanx at a 10 degree angle distally. The first allows for thumb microgestures on the radial side of the index finger, where the finger tip is more directly observable; however, for on-surface tracking, we found the steeper glancing angle of the sensor to the surface to be less sensitive for continuous motion. The second mounting position allows observation of the base of the thumb. During microgestures, this resulted in slightly less consistency of the Δx and Δy values due to human variances of hand position and the relatively smaller motion of base of the thumb to that of the fingertip during microgesturing. On the other hand, mounting the sensor more aligned to the surface enabled more robust continuous motion values. After experimentation, we opted for the latter mounting position because (1) our target microgesture set was discrete and therefore potentially less sensitive to inconsistencies, and (2) the second mounting position permitted a lower profile industrial design.

For the contact microphone, we leveraged Knowles V2S200D,[2] typically used in earbuds in conjunction with traditional microphones for enabling voice clarity during noisy conditions, such as wind noise. We positioned the microphone on the inside of the ring, with the package pressed against the dorsal (top) side of the user's finger. We felt that such positioning on the

[2]https://www.knowles.com/V2S

finger's bonier side could increase the vibration conduction signal from the finger pad to the microphone.

Although a higher SNR could be achieved with an analog microphone and separate amplifier, as in previous works [184, 182], we instead opted for a digital PDM microphone for low power and the lack of additional required circuitry, enabling a fully self-contained design. We tested various decimation ratios between a 4kHz and 16 kHz sample rate while rubbing our fingers and surfaces, and we settled on 4 kHz since it still captured the frequency bands of strongest vibrational power while balancing Bluetooth bandwidth and connection robustness.

Finally, we sample a radially mounted 6-axis IMU (ST LSM6DS3TR-C[3]) at approximately 130 Hz. Each sensor is streamed over Bluetooth LE to a laptop for modeling and inference. Our ring uses two Nordic nRF52840s due to their integrated BLE radio and PDM peripheral controller. A 120 mAh lithium-ion battery powers the device. FlowRing's ability to work completely wirelessly surpasses many prior works, allowing for greater user motion during data collection. It also demonstrates the feasibility of our downstream models to work with real-world challenges of mobile hardware, including increased latency, dropped packets, uneven sampling, and lower SNR sensors. All components are mounted atop a 3D-printed chassis, with an elastic fabric allowing for accommodation of users from ring size of 5 to 8 cm in circumference. The ring weighs 9 grams.

6.4 Input Pipeline

As noted, FlowRing enables two different input methods, 2D continuous on-surface tracking and in-air microgestures. To selectively select between these methods, identify surfaces, and reject false positives from daily activities, we introduce a *gating classifier* (see Fig. 6.3). The following sections describe how we achieve each input method and discuss their performance via user evaluation.

---

[3]https://www.st.com/en/mems-and-sensors/lsm6ds3tr-c.html

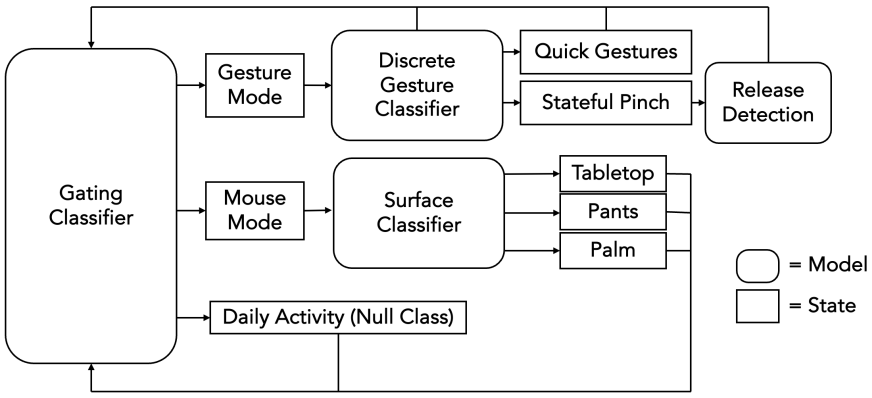


Figure 6.3: *FlowRing's high-level state architecture. A gating classifier rejects false positives from daily tasks and determines if the user performed a gesture or is holding their hand in a mouse-like configuration over a surface; it then routes the samples to the appropriate downstream classifier. If it is a gesture, the discrete gesture classifier then determines the type of gesture, i.e., a quick gesture like a tap or swipe, a longer gesture like a pinch and hold, or a negative sample. If it is a mouse-like action, the gating model can then engage a continuous 2D on-surface tracking scheme as well as trigger surface classification.*

6.4.1 Continuous 2D On-Surface Tracking

Continuous 2D surface tracking requires fast and precise motion, but it also must function across a variety of surfaces, lighting conditions, orientations, and hand shapes. To balance speed and generalizability, we combine both heuristic-based and learned methods. We attempt to emulate the feel of a trackpad to leverage the years of familiarity users have with these types of devices. This task is technically challenging for several reasons: (1) to enable clutching (i.e. lifting the finger or hand to reset new control-display coordinate space), the model must quickly recognize both hand lift off a surface and re-engagement with surface-bound motion, (2) to enable tap selection where the motion sensor is mounted on the finger, the system must quickly recognize finger lift and suppress cursor motion, and (3) the system must support both small, precise finger motions and large display traversals in a limited surface space. The most closely related previous work in this space, LightRing [72], cannot achieve tap selection or clutching since it lacks an accurate measure of whether the user's finger is surface-bound and so cannot replace a mouse.

Method To enable cursor control, we follow the state diagram presented in Figure 6.4. We transition between four states: a latched (1) idle and (2) cursor motion state, and a spurious (3) clutching and (4) tap state. We describe each below:

Clutching Cursor Motion Clutching is an interaction that enables unbounded cursor motion across a large display space. It occurs when the user moves their hand to shift the cursor towards a particular direction, wishes to continue to move the cursor in that direction, but runs out of ergonomic physical range of motion or encounters other physical obstructions to their motion. Enabling clutching is particularly important in our application area for three reasons. First, clutching facilitates cursor control across large field-of-view displays, such as those found in virtual environments, since the ergonomic range of motion of the hand affords a much smaller area than the display. Second, it is important for limited physical spaces, such as an ad hoc control surface like the pant leg. Finally, it enhances ergonomics since the

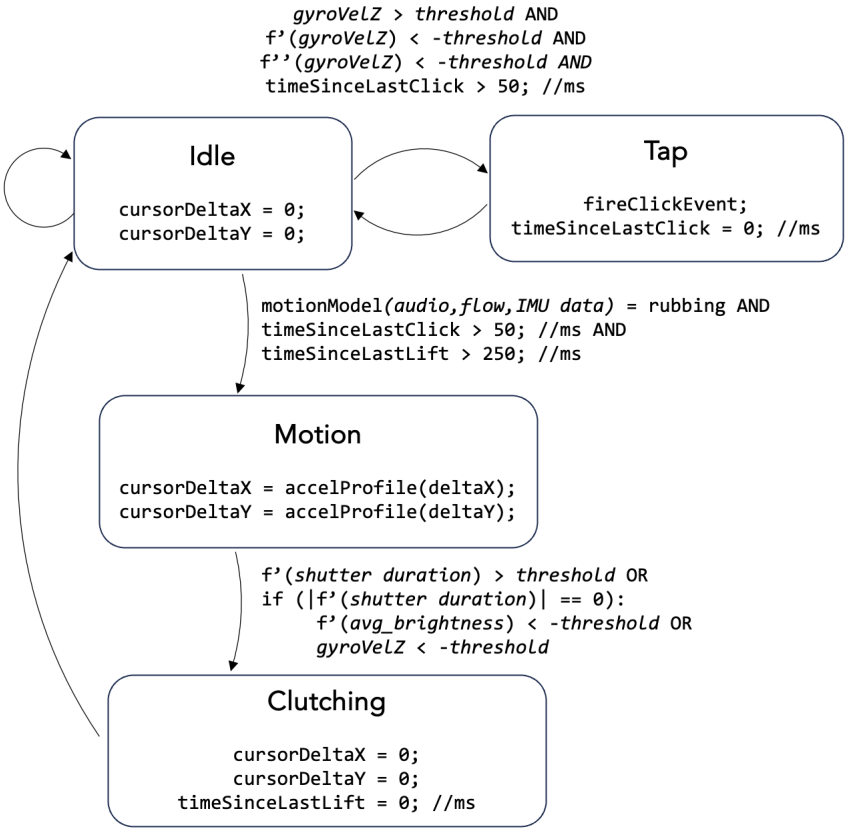


Figure 6.4: *State diagram of 2D surface interaction for cursor control. Transition conditions are listed next to the arrows.*

user can traverse a comparatively large display space with minimal motion of just the finger, wrist, and arm.

To help users apply their previous trackpad usage experience to the cursor control function of FlowRing, we allow users to clutch cursor motion based on whether their index finger is rubbing across any given surface. For real-time usage, a major latency challenge is that users expect perceptually immediate motion of the cursor upon moving their index finger across the surface as well as immediate cessation upon their finger's stopping or lifting from the surface. To achieve precision, users must be able to move the cursor precisely, even with a very slight and slow fingertip motion. For this reason, we chose to adopt a state architecture where we can (1) latch into a *MOTION* state based primarily on a low-latency, high-sensitivity model that detects the start of user finger rubbing, (2) latch out of *MOTION* quickly based on a heuristic, and (3) design an acceleration profile while in a *MOTION* state that suppresses no-motion drift but also enables slow-motion precision. We describe (1) and (2) in this subsection and (3) in the next (Sec.6.4.1).

Entering a Motion State To anticipate whether a user intends to move the cursor, we use the rubbing of the index finger against a surface to clutch. For imperceptible cursor motion, we aim to achieve a max latency of approximately 100 ms total [23, 108] from start of hand motion to cursor motion. Given the latency of BLE packetization and transmission, this leaves a remaining processing budget of approx 65 ms, or approximately 260 samples of audio and 8 samples of flow and IMU data.

We first attempted to use a heuristic model based on the frequency information present in the audio data alone for this task, hypothesizing that rubbing against a surface should create distinct frequency changes. We did indeed see an increase in power in the 400 to 2000 Hz band compared to a hand kept still. However, in practice, we found it hard to separate distinct frequencies, so we instead opted to develop a lightweight, real-time model.

We collected data from 11 participants (see Sec. 6.4.2). Each user rubbed three surfaces within their vicinity — a desk, their own pants, and their own palm — for three 20-second

sessions (remount between each session). Users were instructed to hold their hand in a mouse-like position. This yielded a total of 11 participants $\times$ 3 surfaces $\times$ 0.33 minute $\times$ 3 sessions = 33 total minutes of rubbing data. Each user was instructed to rub at any speed and direction. Since data collection occurred across many different settings, different participants used different desks, pants, and (obviously) own hands. The greatest number of participants who used the same desk was four.

Additionally, we collected negative data on the participants' holding their hand in the air and waving/shaking their hands for 30-second session for 3 sessions with remount, for a total of 11 participants $\times$ 2 actions $\times$ 0.5 minute $\times$ 3 sessions = 33 minutes of negative rubbing data. We also collected 3 sessions of 10 repetitions of each user tapping on the desk to add to the negative data. We trained an SVM that takes as input the fft of 256 samples (64 ms) of audio data. Experimenting with incorporating different sensor inputs did not change the accuracy significantly. In practice, we do not require extremely high model accuracy, rather only high sensitivity, due to latching into a *MOTION* state.

We chose to latch into a *MOTION* state instead of using model output alone to toggle *MOTION*. Model outputs on such short input windows are susceptible to spurious misclassifications. By latching into this state, we could also avoid additional latency associated with majority voting a sliding window across the model's output. We could also enable very precise per-pixel cursor control when the finger was moving very slowly since we could avoid fluctuating model output given minimal or no IMU or audio signal. By latching, this becomes a non-issue.

Exiting a Motion State In our experimentation, we found exiting the motion state to have even stricter latency requirements than entering. This is due to high user perceptibly of cursor motion immediately prior to selection. As the user lifts their index finger to start a selection tap event, the cursor must immediately decouple from finger motion to avoid moving the cursor off a small target, for example. Therefore, to exit the motion state, we employ a low-latency heuristic on optical and IMU data to detect lift off the surface.

Specifically, we notice that the reflectivity of a given surface remains approximately constant as the user moves their hand for cursor control. The finger or hand lift results in a sudden change in the returned signal power to the optical flow sensor. This is most clearly seen by an increase in *shutter duration*. Therefore, we threshold the derivative of the most recent 4 samples (approx 30 ms) to determine lift. However, for users with large hands, where the sensor sits far from the surface, or for low reflectivity surfaces, we noticed that the shutter duration remained pinned to its maximum value. In that case, we threshold for a decrease on the derivative of *average frame brightness*.

A potential failure mode may be an unintended motion state exit when moving across a highly variegated surface (e.g., a checkerboard surface). However, in practice, we did not encounter that issue even on patterned desks or on multi-colored paper prints. A final signal of user finger lift is high rotational velocity around the radial-ulnar axis (`gyroVelZ`). Any one of the preceding criteria triggers a fast decoupling of motion from movement.

During Cursor Motion Once the cursor is in motion, we apply an acceleration profile to balance speed and precision. Slow hand motions result in small cursor displacements; quicker hand motions result in disproportionately larger cursor displacements. Acceleration is applied on a per time-step basis. Thus, given $\mathbf{u} = \langle \Delta x, \Delta y \rangle$, the accelerated cursor displacement is defined by the equation

$$\mathbf{u}_{accel} = \mathbf{max}(\mathbf{round}(0.13 \times \|\mathbf{u}\|^2), 300) \cdot \hat{\mathbf{u}}. \tag{6.1}$$

By applying this acceleration profile and rounding, the sensor's Δx and Δy noise are below 0.5 with no motion and therefore drop to zero after rounding. We constrain maximum cursor speed (acceleration multiplier = 0) for very high mouse motion to limit the user's losing their cursor on the screen. We apply the acceleration profile equally in any direction. Even if the sensor is slightly tilted with respect to the surface, we find that it does not change user perception.

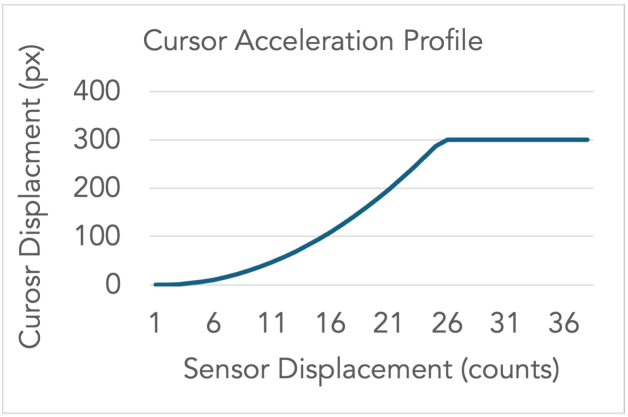


Figure 6.5: *Acceleration profile applied to cursor movement.*

Selection Finally, we use characteristics of the rotational velocity of the index finger to generate tap events. Specifically, we threshold on high rotational velocity around the radial-ulnar toward the palm, high rotational deceleration, and high rotational jerk. These features are present in the `gyroVelZ` signal when the finger strikes a surface.

We first prototyped heuristics around contact microphone tap events. However, we found finger motion to be a more robust heuristic across different textures, such as soft pants and hands as well as hard desks and walls. To prevent multiple tap events from triggering sequentially, we apply a cool down period of 50 ms. This still enables intentful double clicks that exceed 150 ms between two clicks.

Performance Evaluation To measure the performance of our pointing and selection system, we invited ten participants to try our system in a 2D Fitts'-style evaluation.

Participants and Apparatus Participants' (2F, 8M) index finger ring circumference ranged from 6 to 7 cm ($\mu = 6.5$ cm, $\sigma = 0.46$ cm) and hand lengths (palm to middle finger tip) ranged from 17 cm to 20 cm ($\mu = 18.4$, $\sigma = 1.29$ cm). All participants had never previously used FlowRing for cursor control.

Experiment We tested FlowRing in three conditions, i.e., on a desk surface , on the participants' own pant legs, and own palm. We use Wobbrock's FittsStudy application [173] to run an ISO 9241-9 compliant study on an 27-inch Dell Monitor U2719DC (screen resolution 2560 × 1440 pixels, 60 Hz refresh rate), where the system cursor is controlled using the preceding mouse algorithms running in real-time on an Alienware 17 R4 laptop.

Users performed six *Amplitude (A)* × *Width (W)* conditions with 2 levels of W (50 and 150 pixels = 1.1 and 3.3 cm) and 3 levels of A (250, 350, and 450 pixels = 5.7, 8.0, 10.2 cm) to produce nominal *Indices of Difficulty (ID)* of 1.415, 1.737, 2.0, 2.585, 3.0, 3.3219 bits (though univariate Crossman-corrected IDe's were used for throughput calculation). Participants were instructed to wear the ring on their right index finger, were shown the hand position for sliding and clicking, and were given a set of practice targets until they felt comfortable. We note that participants quickly adapted to FlowRing's interaction by applying their similar experiences with mouse or trackpad-like motion. No participant needed more than one minute before declaring they were ready to commence the trials. We collected one $A \times W$ block on each surface. The order of *ID* conditions was randomized, and the surface was counter-balanced.

Because our ring uniquely enables pointing *and* select functions, and due to the unavailability of previous pointing-only hardware like LightRing [72], there is no comparable state-of-the-art device to baseline against in a straightforward manner. We also collected as an upper-end an additional session on the laptop's built-in trackpad, despite its being a stationary and low-latency wired device. Due to the different affordances between our wearable and the trackpad, we provide it as a comparative data point rather than as a baseline condition to beat to allow researchers to calibrate future cross-study findings.[4]

Results All participants successfully completed all conditions, indicating that FlowRing can enable surface interaction in a user-independent manner. The average movement and

[4]Due to the difficulty of reproducing hardware prototype systems, we encourage future researchers to do the same so results can be compared.

selection time for a desk is 1637 ms ($\sigma = 283$), for pants is 1603 ms ($\sigma = 353$), and for palm is 2153 ms ($\sigma = 543$) (Fig. 6.6, center). Participants felt the system was quick and intuitive on desk and pants; a sense of system performance can be best understood from this video figure: `https://youtu.be/Dj4L9RbjL4c`. However, performance was comparatively worse on the palm due to its small tracking size and noisier IR reflectivity. The palm therefore would likely be more appropriate for 2D touchscreen like manipulations rather than cursor control across a large screen. Two participants reported that given the roughness of the desk we used and the stickiness of their fingertip from sweat, they experienced friction, especially during upwards movements. All participants could clutch without issue. Most users were able to tap without issue. Two participants occasionally needed to tap multiple times on the desk to trigger selection; after being reminded to tap swiftly, they had no issue, and their data was retained. However, before correction, their cursor movement was more erratic during touchdown and thus stood as error rate outliers. Error rates were 6.25%, 4.58%, 9.52% for desk, pants, and palm respectively (Fig. 6.6, center).

We calculate throughput via a mean-of-means method [144], finding good agreement across users. The average TP_{avg} across users for the desk condition was 1.54 bits/sec ($\sigma = 0.23$), for pants was 1.64 bits/sec ($\sigma = 0.27$), and for palm was 0.86 bits/sec ($\sigma = 0.21$) (Fig. 6.6, left). Between desk and pants, a paired t-test indicated no significance ($p = 0.10$) as well as no correlation across participants. This result was moderately surprising since we expected the curved and soft nature of the pant leg to make cursor control more challenging. However, this result indicates the potential of FlowRing to work on non-traditional, ad hoc surfaces. For reference, the TP_{avg} of the built-in trackpad was 2.66 ($\sigma = 0.38$), and the movement and selection time was 1101 ms ($\sigma = 181$).

6.4.2 Discrete Gesture Classification

For the smallest surface of fingertip our system detects thumb-to-index microgestures. To control a simple, smartglasses-type UI. Our gesture set consists of five quick gestures: left swipe, right swipe, up swipe, down swipe and tap, and one stateful gesture, pinch.

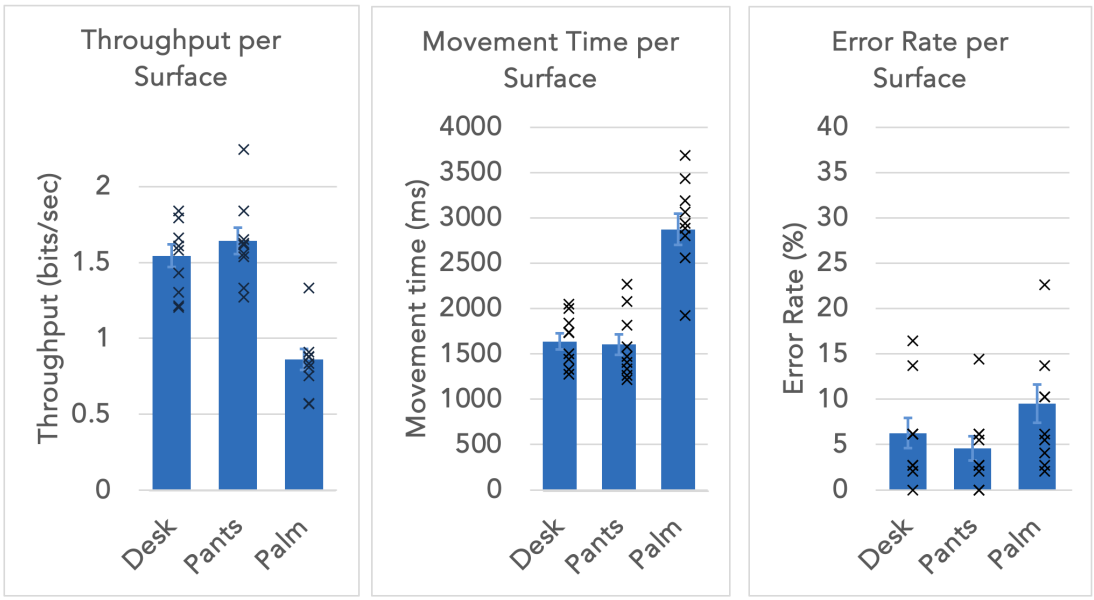


Figure 6.6: *Results from the Fitts' pointing and selection study. The larger desk and pants surface performed similarly, the smaller palm surface is slower for the same pointing task, potentially due to smaller control area. Error bars represent standard error. Marks represent per participant data.*

Method We detect these five quick gestures and the onset of pinch by a neural network consisting of a 3-layer CNN and an LSTM. For the stateful pinch, an SVM model detects the release of the pinch. To improve the accuracy of detection, we fused data from a contact microphone, an optical flow sensor and a 6-DOF IMU. To evaluate, we performed a leave-one-out experiment, fine-tuning for new users, as well as an ablation study. We describe each in the following sections:

Data Collection and Segmentation We recruited 13 participants for data collection, however retained data from 11 (5F, 6M) due to significant data loss during streaming in two instances. Their average hand size was 17.94 cm, average ring size was 6.25 cm, and their Fitzpatrick skin tone ranged from I to IV. Each participant spent around 1 hour on

data collection activity, which was divided into three sessions. Each session consisted of approximately 5 minutes of discrete gestures data, 5 minutes of negative data, and 10 minutes of on-surface interaction data (see Section 5 for details). In between sessions, the participants were required to remove the ring and re-wear it to simulate different wearing situations. The negative data contains some aggressor actions in everyday life (hard negative: writing, clapping, typing, using the phone, hand clenching, and waving hand freely) and data when the hand is at rest (soft negative).

Because each microgesture is short and the duration of the action differs for each participant, segmenting positive samples is challenging. To solve this problem, we built a data collection game similar in style to "Dance Dance Revolution" with sliding prompt boxes. The participants performed the gesture within a 1.2 second "keep-in" sliding box (green box in Fig. 6.7), and centered their gesture on a short, 0.2 second prompt (red box in Fig. 6.7) within that larger box. This approach provided a couple benefits: as we strided our 1-second model input window (blue box in Fig. 6.7) across the “keep-in" interval, each observation was guaranteed to contain true positive gesture data, and this striding effectively provided time-shift augmented data.

Data Preprocessing We aligned the three types of sensor data by timestamps and segmented the data into samples with a sliding window of 25 millisecond stride. After that, we performed feature extraction on each sample.

Fig. 6.8 shows an example of the raw data. First, we applied FFT (length of 512, overlap of 64) on the audio data, which produced a $n_{freq} * n_t$ matrix, where n_t represents a microgesture that consists of n_t time sequences. For 5 types of optic flow data (Δx, Δy, image keypoints quality, shutter duration, average brightness) and 6 types of IMU data (6-DOF), we first interpolated the data to match lengths and used the average filter for noise reduction. Then, we reshaped each type of data into a $n_{seq} * n_t$ and concatenated them with the audio feature.

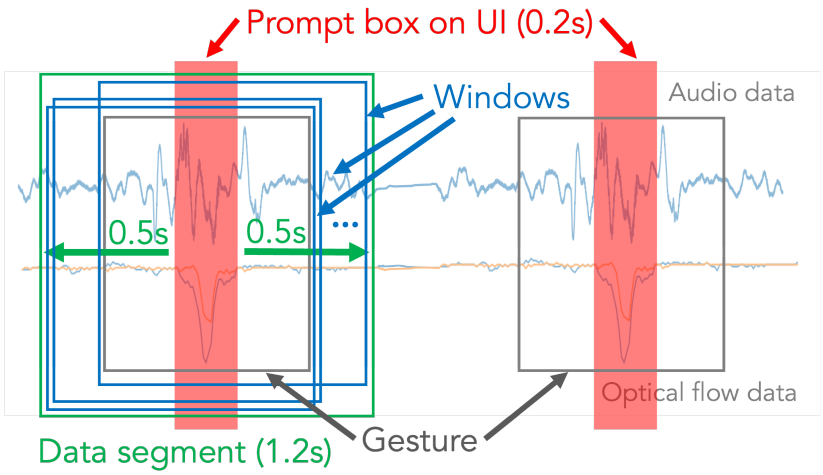


Figure 6.7: *We showed a prompt box on the user interface (UI) at the middle of each gesture when collecting data. When the participant makes a gesture, the prompt box should be located in the middle of the gesture to the extent possible. Each data segment (green box) contains a prompt box of 0.2 seconds plus 0.5 seconds before and after it. Then, the sliding window is applied to generate data samples from segments. The data samples (blue boxes) include the valid data from the gesture.*

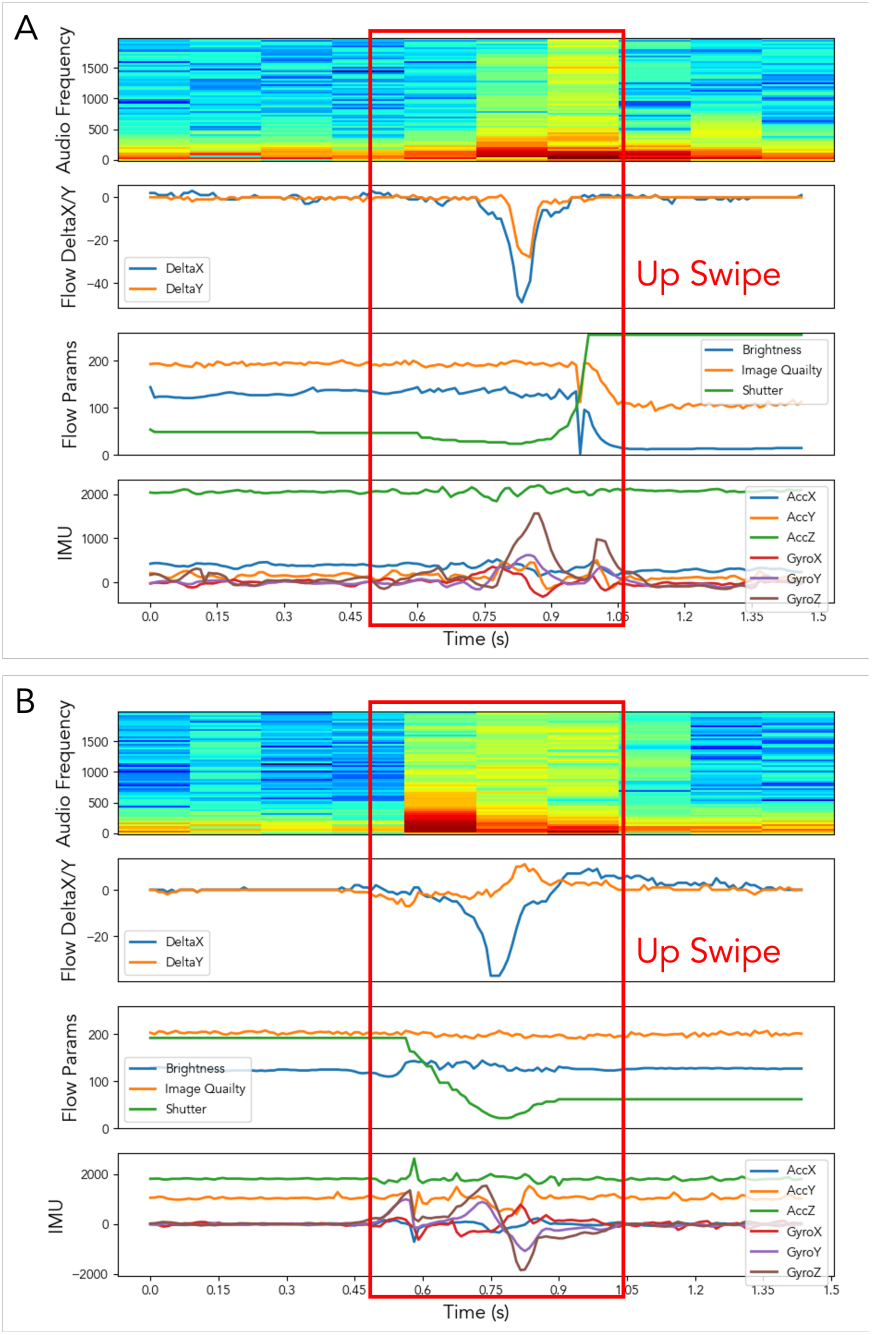


Figure 6.8: *The raw data from three types of sensors and the spectrogram of audio data. A: An up swipe performed by a participant with small hands. B: An up swipe performed by a participant with large hands. The amplitudes of their optical flow data differ, providing an example of an aspect our model needs to generalize across.*

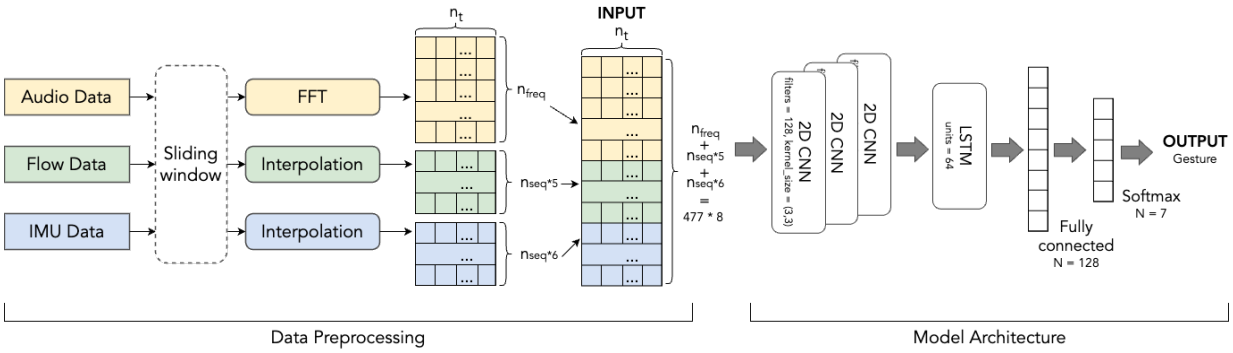


Figure 6.9: *Model input consists of features extracted from three sensing modalities (left). The quick gesture classification model contains three 2D CNN layers followed by a MaxPooling layer and an LSTM layer. The fully connected layer and softmax are then applied (right).*

Quick Gesture Classification Model The main part of the model is a three-layer CNN network and an LSTM network. The rationale for these choices is that 2D CNNs perform well when extracting features from 2D images, while LSTMs can capture temporal relationships in the sequence data. Each CNN layer has a depth of 128 and a kernel size of 3×3. This is followed by a maxpooling layer with a pool size of 1×2, where the size on the temporal channel is 1 to avoid losing the temporal information. We also added a dropout layer and a regularizer to prevent overfitting.

During training, we used a dynamic learning rate starting at 1e-5 and added early stopping to prevent overfitting. We used the sparse categorical cross-entropy as the loss. The model was trained on an RTX 4090 graphics card.

Stateful Pinch Detection The fast gesture classification model detects the onset of a pinch. After it is detected, the state of the pinch is entered. Then, we use an SVM (C=1, kernel=rbf) model to detect the release of the finger, which is also the end of this pinch.

Performance Evaluation The fast gesture classification model classifies the gestures into 7 categories: left swipe, right swipe, up swipe, down swipe, tap, the onset of a stateful pinch, and none of the above. Below, we first report the main gesture classification results of our model. Then, to investigate the potential of one-time calibration, we report results of a separate fine-tuning experiment we performed to measure the model's ability to learn new users' data, during which a small amount of data from a new user was added to the training set. Finally, we show results from an ablation study to observe the effects of different sensing modalities of data on the results.

Gesture Classification Results For our main result, we generate three accuracies: within-session, leave-one-session-out (LOSO) and leave-one-user-out (LOUO). For the within-session setting, we mixed data from all users and all sessions and divided it into the training set and the testing set according to 80:20. In the LOSO setting, we used the data from two sessions for training and from the other session for testing. We repeated for all three sessions and calculated the average accuracy. In the LOUO setting, we used the data from ten participants for training and tested on the data from the left-out participant. We repeated this 11 times for each participant and calculated the average accuracy.

Table 6.1 shows the main results. The accuracy of the within-session setting is 93.86%, which shows that our model can learn the characteristics of different discrete gestures well. The LOSO result is slightly lower than the within-session setting, with an accuracy of 93.61% demonstrating that our model can also work well across sessions and may not need the per-session calibration. For LOUO, the average accuracy is 85.22%, with the highest accuracy of 96.75% (P2) and the lowest of 76.14% (note that P10 has the largest hand among participants), as shown in Figure 6.11. Hand sizes and finger lengths vary across users, which may result in differences in the same gesture. For instance, as shown in Fig. 6.8, the data of participants with small hands does not change in Δy as much as that of users with large hands when performing an up swipe.

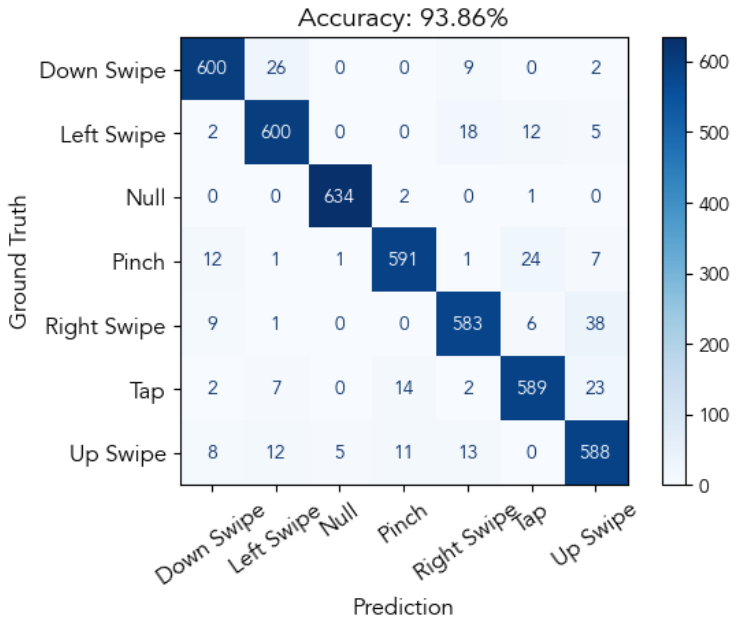


Figure 6.10: *The confusion matrix of the quick gesture classification model.*

Limited Fine-Tuning for New Users Rings are a personal device sized to a single user, therefore similar to FaceID or pHRTF ear scan for spatial audio, we are interested in exploring the potential of a minimal, one-time calibration for boosting accuracy. To investigate whether our model can learn the features of a new user's gestures from very few samples, we added a small amount of data from the new participant to the training set when doing LOUO. We used 10 participants' data from all three sessions plus a small amount of data from the left-out participant's first sessions for training and then tested on the data from remaining two sessions of this participant. We did the same process for each user and calculated the average accuracy. In two experiments, we randomly chose two repetitions and four repetitions of each gesture and added them to the training set. The results in Table6.2 show that the accuracy of the two experiments improved by 3.32% and 4.93%, respectively, indicating that a very limited one-time calibration scheme can yield high accuracy even across remounts.

Table 6.1: *Main results of the quick gesture classification model.*

	Within-Session	LOSO	LOUO
Accuracy	93.86	93.61	85.22
F1-score	93.86	93.59	84.84

Table 6.2: *The results of minimal new user calibration. Accuracy for leave one user out, inclusion of two gesture set examples from a new user, and of four.*

	LOUO	LOUO +2 repetitions	LOUO +4 repetitions
Accuracy	85.22	88.54	90.15
F1-score	84.84	88.41	90.01

Ablation Study As shown in Table 6.3, we performed an ablation study to identify the contribution of each modality by removing the features extracted from each. We compared the results to the within-session setting.

After removing the audio features, accuracy drops marginally. This confirms our assumption that the audio data obtained from the contact microphone can help detect rubbing but is either not very useful for distinguishing between different gestures or is partially redundant to the IMU. The results after removing optical flow and IMU data also show that they contribute more to gesture classification, capturing motion as well as rubbing. As shown in Fig.6.8, there is a clear signal when the gesture is performed.

Result of Stateful Pinch Model The negative data of this setting is the data when the hand is keeping still, and the positive data is the release data. We trained a simple SVM on the data from the first two sessions and tested it on data from the last session. The accuracy of our pinch release model is 94.2%. Since this is gated by the gesture classification model,

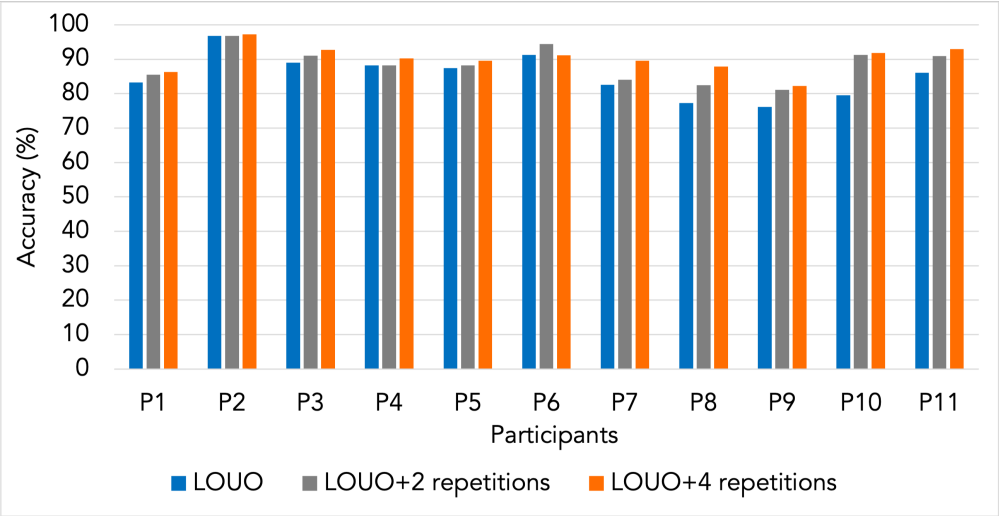


Figure 6.11: *The per-participant accuracy of the quick gesture classification model after adding two or four repetitions of the gesture set.*

Table 6.3: *Results of the ablation study.*

	Within-Session	w/o Audio	w/o Optic flow	w/o IMU
Accuracy	93.86	92.24	86.27	87.62
F1-score	93.86	92.23	86.32	87.63

we found that an inexpensive release model is sufficient for real-world use.

6.4.3 Gating and Surface Model

Method

Gating Model Given that our system offers two modes of input—in-air and on-surface—we built a gating model to discern and respond to the user's specific interaction. This model processes input data streams from all sensors, including optical flow, acoustic, and IMU

data, to categorize the input into three distinct categories: in-air input, on-surface input, or no input. The purpose of this model is twofold: it aids in identifying the user's intended interaction and effectively filters out false positives when no participant interaction is detected.

To train this model, we repurpose the data collected from discrete gesture studies and continuous input experiments. We consolidate individual gesture classes into a unified *gesture mode* class, merging actions such as tapping, swiping, and others. Additionally, we combine instances of rubbing on pants, hands, and tabletop into a collective *continuous mode* class. Finally, we group non-input-related activities, such as writing, 'clapping, typing, phoning, and clenching, into a single "null" class to serve as hard negative data.

The gating model's architecture and training methodology closely mirror those used to train the discrete gesture classifier model. See Section 6.4.2 for in-depth information.

Surface Model To support seamless cross-surface interaction, we built a surface model to distinguish the type of surface the participant interacted with. The model's input is the same as the gating model to reduce the number of data processing steps. The model's output is the three surfaces, "tabletop", "pants" and "palm", as well as "null", which is composed of soft and hard negative data (e.g., "typing" and "writing"). The fourth surface, fingertip, is returned if the gating model detects *gesture mode*. The architecture and training methodology are also similar to those used for the gating model.

Performance Evaluation The evaluation method and dataset split are similar to the evaluation for discrete gesture classification (see Section 6.4.2).

The accuracy of the gating model under the within-session setting 99.21%. Results of LOUO and LOSO are slightly less accurate, but both exceed ninety percent (see Table 6.4). This shows that our gating model works well across users and sessions in real-world scenarios.

The accuracy of the surface model under the within-session setting is 97.97%, and the result of LOSO is slightly less accurate. However, the accuracy of LOUO drops to only 85.76%, perhaps due to the variety in colors and materials of participants' pants and the

colors and textures of their palms.

Table 6.4: *Results of the gating and surface model performance.*

	Within-Session	LOSO	LOUO
Gating model accuracy	99.21	97.30	93.82
Gating model F1-score	99.21	97.30	93.83
Surface model accuracy	97.97	94.70	85.76
Surface model F1-score	97.97	94.71	85.25

6.5 Discussion

6.5.1 Applications

FlowRing's interaction space extends seamlessly across surface sizes. This continuum (see Fig. 6.1) can be mapped to support applications that span multiple dimensions, such as different postures, different UI affordances, and different devices. We build and demonstrate these applications in the accompanying Video Figure.

Cross-Posture Control As users leverage a given computing device in different scenarios, their posture may change with use. For example, a user wearing an XR device may start at their desk. Since they have the greatest level of expressivity and precision on this surface, they may use precise pointer control with an application (e.g., Spotify's music player) in a desktop version, with high visual and target density. If they were to move to their couch, FlowRing would detect the user's thigh as the available interaction surface. The new surface would still allow for cursor control, but perhaps with slightly less precision than a perfectly flat, hard and smooth desktop surface. The headset could then show them a scaled Spotify UI with larger target elements and a cursor (e.g., like Apple iPad Pro's cursor experience). Finally, if they were to take a walk, the app might shrink to provide only key controls and

information, such as a music miniplayer. To navigate the limited elements of the miniplayer, the individual could use discrete microgestures to select amongst the few interactables (Fig. 6.12).

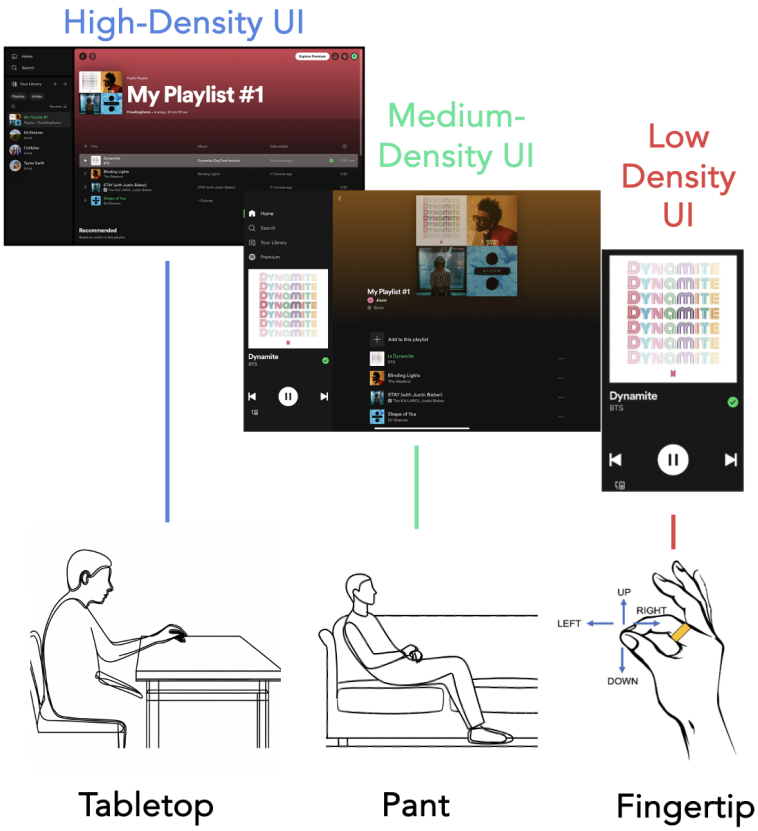


Figure 6.12: *FlowRing enables seamless input across different user postures. As FlowRing recognizes different surfaces, the user wearing an XR headset would be provided a desktop-like, high-density UI while seated at a desk with high precision, a tablet-like, medium-density UI while seated on the couch with gestures on pant, and a mobile-like, low-density UI while on the go with fingertip microgestures.*

Cross-UI Affordance Control Within an application or digital environments, multiple controls can be mapped to multiple surfaces based on their affordances. For example, for a

VR whiteboarding application, a user could draw on the canvas with a pencil tool. A 2D color circle picker could then let them select their desired inking color. Further, tool modes, such as eraser, line, circle, etc., could be available in a tool menu. To couple the UI elements to surfaces based on affordances, control of the canvas inking could be mapped to the largest and most precise surface: the tabletop. The ad hoc palm surface could enable a shortcut for quick color adjustments with finger drags. Finally, discrete swiping microgestures could provide a shortcut for quickly switching tools (Fig. 6.13).

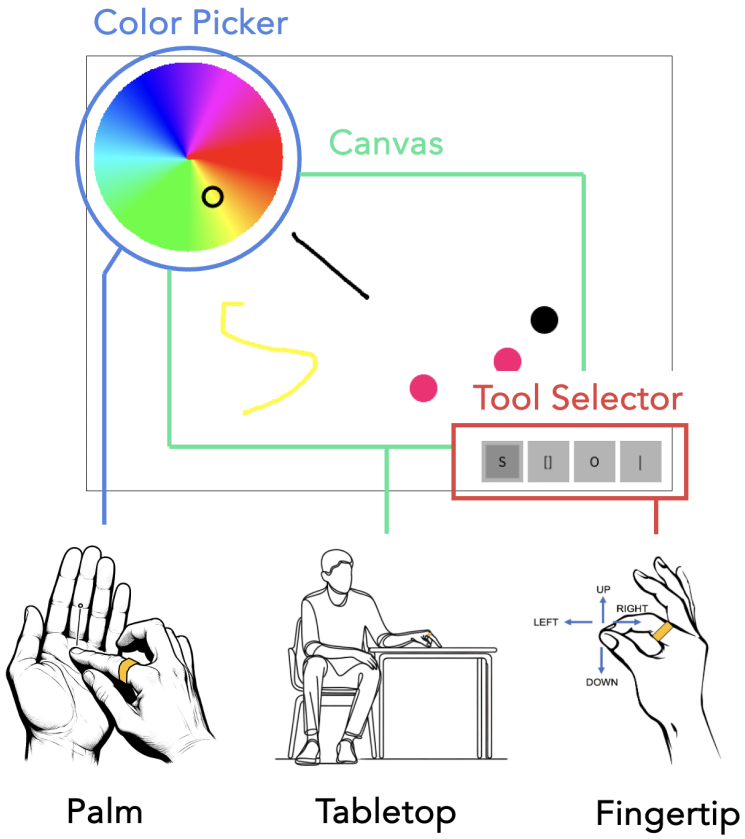


Figure 6.13: *UI elements can be mapped to surfaces according to affordance. FlowRing allows for color to be selected on the palm, the canvas to manipulated via tabletop swipes, and tools to be selected with fingertip microgestures.*

Cross-Device Control Users now commonly interact with multiple devices concurrently. With FlowRing, different devices can be associated with interactions on different surfaces. For example, users could use a mouse-like pointer on the desk surface to interact with their desktop VR or computer. Concurrently, they could control their connected earbuds with specific microgesture controls (swipe left/right for prev/next song, up/down for volume control, tap for play/pause) and adjust their environment by linking swipes on the palm with the intensity and warmth of their smart lights (Fig. 6.14).

6.5.2 Future Work

We look toward further characterizing our device, addressing limitations and improving our design in future iterations.

Hardware Power and Size Optimization Though our prototype device demonstrates completely wireless functionality, there is room for improvement. For ease of maintaining bandwidth across various pilot experiments, our prototype used two microcontrollers and radios, but a future implementation could easily use one. Despite this, our ring consumed only 91 mW when constantly streaming data, or approximately 5 hours on the 120 mAh battery. This power consumption includes two MCUs, both power supplies for each, two BLE active radios, the optical flow chip and associated regulators, the contact microphone and the IMU. Power and form factor could be improved by consolidating these components. Unlike other optical-based devices [75, 163] that may require line-of-sight to the fingertips for function, we instead design FlowRing to keep sensors in a low-profile orientation. Fig. 6.15 shows an example of what a future system could look like with a custom flexible printed circuit (FPC).

Further power savings could be realized by duty-cycling the ring to wake only on distinct triggers, like a unique accelerometer-based double tap or an incoming message on a head-worn display. Finally, investigating quantizing and shrinking the gating model to run on-device would allow the radios to remain off the vast majority of the time, waking only when needed.

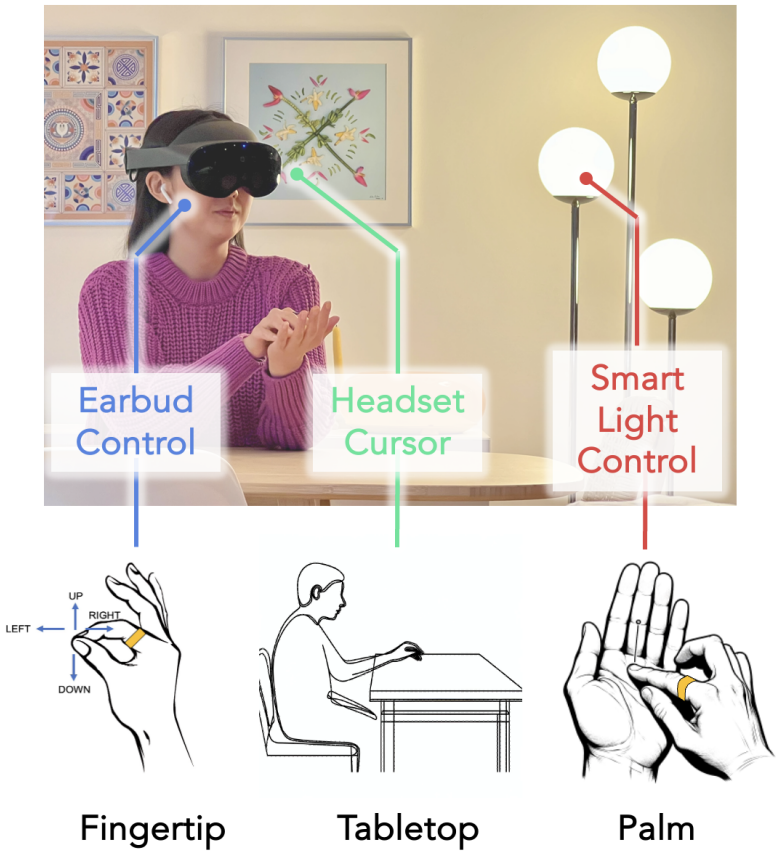


Figure 6.14: *Multiple devices can be associated with specific surfaces to enable seamless cross-device control. The intensity and warmth of the smart lighting can be controlled with palm swipes, the earbuds' music functions can be controlled via fingertip microgestures, and the headset's cursor can be controlled on the tabletop.*

Figure 6.15: *Mock-up of a future FlowRing with a custom flex PCB.*

Identifying Learning Effects and Fatigue Effects Within our Fitts' evaluation, we performed only a single block of trials for each condition. It would be interesting to extend this evaluation to a longer time period to identify if users' performance would improve as they grew accustomed to the device or deteriorate if fatigue or other effects, such as sweaty fingertips, became a factor.

Longitudinal Study We provide a first investigation and concept for a multi-surface ring. To allow the system to generalize to truly real-world use, a longitudinal, in-the-wild data collection and study would allow for a greater number and variety of observations across re-wears, surface types, motions, ambient lighting, and demographics. The fact that our ring is wireless lends itself toward such a data collection, however more engineering would need to be done to make the logging platform itself truly mobile.

3D Interaction Techniques Spatial computing devices also feature 3D interactions and manipulations. While we do not specifically investigate such interactions techniques in this chapter, FlowRing has the potential to allow for 3D manipulations with IMU + pinch detection. It also can be potentially used for on-surface 3D manipulation by using touchdown + finger yaw interaction techniques, such as those proposed in 3DTouch [112].

6.6 Conclusion

This chapter introduced FlowRing, a novel ring wearable designed to cater to the varied input needs of both on-the-move and immersive desktop contexts. Our investigation demonstrated that FlowRing, which features an optical flow sensor, skin-contact microphone, and IMU, can facilitate versatile input across tabletop, pants, palm, and fingertip. User studies substantiated its potential, showcasing promising gesture recognition accuracy across sessions and users. Furthermore, a successful 2D Fitts law test underscored the system's effectiveness in handling continuous input tasks. FlowRing represents a significant step forward in enhancing mixed reality interactions and holds promise for further advancements in this field.

Chapter 7

CONCLUSION

My thesis proposes that *by implicitly leveraging the spatial context of a user's environment, technologies can better serve users as they operate within a situation, for example, while participating in another task (Chapters 3 and 4) or operating while mobile (Chapters 5 and 6).*

In Chapters 3 and 4, we investigate *spatial computing interaction* through building a system to support electrical engineers in debugging PCBs. For this task-oriented system, we find that spatial computing can provide value by reducing context-switching and spatial pattern matching within the user's traditional workflows, but that it must also exhibit stability and precision of the smallest salient element for the full value to be realized.

In Chapter 5, we explore *spatial computing sensing* by building a set of spatial audio capture earbuds. We develop a novel pipeline that leverages both spatial and acoustic information for real-time, mobile speech enhancement. This demonstrates the utility of a system that can fuse spatial context with other types of context to outperform traditional systems that use just spatial information (beamforming) or just acoustic information (mono-channel learning techniques) alone. We specifically discuss challenges in deploying in-context, including developing custom hardware, making our model perform in real-time, and training with real data.

Finally, in Chapter 6, we develop a system for *spatial computing control* that can work across stationary and mobile contexts. We build a optoacoustic ring peripheral that allows for surface-bound interaction across a range of always-available surfaces, including precision cursor control and quick fingertip microgestures. We work towards satisfying multiple design constraints needed for practical, in-context use by opting for low-power sensors, a minimal,

one-time user calibration, robustness across re-wears, and a low-profile location and the base of the index finger.

7.1 Lessons Learned

Generalizing the conclusions from our works produces several design considerations for the future of *spatial, in-context computing*. To deliver the most immediate value to users, one should focus on augmenting current user journeys and workflows. It is critical to identify and address points of frequent physical-digital context switching, spatial pattern matching, and high friction of information retrieval. Providing just-in-time and spatialized information is the defining opportunity for in-context spatial computers as it aligns with user's intuition and faculties of operating within a spatial world. However, for the same reason, the delivery of information is also subject to a high bar of user expectations, especially that the system's perceptive and augmentation capabilities match their own. As such, there is no shortage of engineering challenges to deliver the sensing and perception systems for in-context computing. In addition to challenges of traditional interactive systems such as low-latency and low-power operation, in-context systems need to generalize sensing and interaction techniques to operate across multiple environments, scenarios, and contexts.

From a practical standpoint, this final point has underscored the critical importance of a basic principle in system building, one that scales from circuit design to building organizations: constructing a complex system necessitates reliable subcomponents. While I first learned via my electrical engineering background, I find it is even more crucial in the context of research, where the path from concept to creation is more circuitous and ambiguous.

Despite sounding obvious, it is a good guiding principle for modern interaction design as well, where techniques are increasingly complex. For example, applying this concept to learning-based processing highlights the importance of generating a clean and clear understanding of the raw data, including visualizations. It also motivates the training of models in simpler, more controlled situations before tackling more complex environments.

Similarly, when initiating a project, removing design constraints and testing the hypothesis

in isolation can quickly validate the approach or reveal potential flaws. With a bit of generalization, this tenet can also be abstracted to larger endeavours such as a research or product planning. For example, working towards a "north star" research or product concept can involve generating stepping stones along the way that can each provide user value, before building towards a larger goal (e.g. ClearBuds with two wireless time-synced nodes is a subset case of an ad-hoc multi-microphone array for conference rooms which has a room-scale number of nodes).

7.2 Towards Ubiquitous, Spatial Computing in Context

As we look beyond smartphones, the next chapter of ubiquitous computing may be spatial. Beyond my dissertation, I hope to continue this research thread to build practical, efficient systems for mobile spatial computing input. One form factor that has garnered much interest from industry and academia alike is smartglasses. To realize the potential of smartglasses and other spatial computing platforms, advancements are needed throughout the human-computer interaction stack: in engineering resource-efficient sensors and perception algorithms, in designing precise, ergonomic, and socially acceptable interaction techniques, and in developing interaction models that align with user intention, acting as "an helpful, invisible servant" [166].

7.2.1 Lightweight Sensing and Interface Technology

For spatial computing to robustly perceive the world in real-time, sensors and algorithms must efficiently process vast amounts of environmental data. The technological forefront of devices today, such as Meta Quest Pro and Apple Vision Pro, favor camera-based methods, piggybacking on the immense improvements in power, integration, and cost of smartphone CMOS image sensors. Particularly in the context of smartglasses, optical and other sensing modalities will need to continue to make strides on size, weight, and power (SWAP)[1], especially

[1]Let alone the work ahead for display technologies on SWAP as well. [82]

for ambient compute. Interesting approaches include low-framerate, low-resource, ML-based gating models that run on the image sensor co-processors to avoid waking the application processor (e.g. hand detection model gating a more costly gesture recognition model). Other approaches include offloading sensing and compute to other devices, such as the approach taken with FlowRing (placing sensors on a separate ring instead of headset device) and ClearBuds (leveraging the smartphone for deep learning compute).

In designing for in-context use, it is crucial for these sensing technologies to be context-sensitive – this includes ensuring they meet social norms and protect user privacy. In recent years, we have witnessed significant shifts in societal and generational attitudes toward technology. For instance, the front-facing camera of Google Glass faced severe criticism in 2014, yet a similar feature became the main feature of Snapchat Spectacles in 2016 and Ray-Ban Stories in 2021. Furthermore, always-on microphones in smart speakers are now found in over 35% of U.S. homes [87], demonstrating a substantial trust from users that their data will not be misused. Emphasizing lightweight, local processing methods can safeguard this trust by confining sensitive data to the user's own device. Looking forward, I forsee a future where ambient, always-on sensing connected to an obfuscated processing network becomes an accepted social norm, with little expectation for direct user interaction. However, in the near term, data that is both interpretable and accessible to humans will remain particularly delicate, necessitating clear and immediate feedback to both users and, when relevant, bystanders.

7.2.2 Multimodal Input and Interaction

For both sensing and action, human rarely rely on a single sensing modality. Identifying complementary modes that align with users' natural behaviors can allow for greater input comfort and throughput. For example, Gaze + Gesture [31] combined the implicit pointing aspect of eye gaze with the explicit manipulation of gesture to yield a rapid targeting and selection mechanism, as seen in Apple Vision Pro. As we consider in-context use, multimodality may become even more crucial. Take, for example, a visual query scenario

where a user asks their smartglass assistant the price of a bottle of wine in their hand. Such a scenario combines a voice query with an optical image of the scene. Further, the image region of interest may defined by their gaze region or by identifying hand key points to isolate the held object. Dynamic contexts further motivate multimodality – for example, voice may be appropriate for certain scenarios and not others, prompting the user to another modality for text entry. To support such interactions, in-context interaction models will need to parse multiple sensor input streams, execute tight hand off between multiple modalities, and define the semantic relationships between actions across multiple input modalities.

7.2.3 User Intention beyond Input

Interactive system design attempts to create the highest bandwidth channel between user intention and device response. Traditional spatial interaction models, such as controllers or hand raycast, have leveraged purely explicit user actions. Gaze and gesture take a half step toward a more implicit model by hybridizing explicit hand gestures with gaze targeting, an implicit indication of the user's intention. (To truly fulfill this description, gaze would be processed in a way that is not merely indicative of the direction of the user's eyes, but by also monitoring fixation events and smoothing between microsaccades to better align with user's locus of attention.)

By further unlocking multimodal input streams, future interaction models can increasingly incorporate implicit user queues. Rather than the direct input $\rightarrow$ state change paradigm of previous computing generations, a paradigm of multi-modal input streams $\rightarrow$ user intention $\rightarrow$ state change could allow spatial computing to more seamlessly and intelligently support the user. Achieving this model opens multiple research questions: What is the mapping from multi-modal input streams to user intention? Can it be informed by heuristics (e.g. when a user faces an object they intend to interact with it) or learned from user behavior data? What is the best way to represent intention given its probabilistic nature? At what point does the system trigger a state change given this information? As input is the gateway for computing systems, an interaction system that it not reliable can result in immense user frustration.

While leveraging implicit data can enhance user experience, it also risks frustration when inaccuracies occur.

In future work, I hope to create an "intention framework" that collects and analyzes various sensor data, trialing different ways to represent user intention (for example, generating a saliency heatmap of the user's visual field). Given the vastness of the input and output space, such a framework may need to be underpinned by foundation models. We already see multimodal vision and language models facilitating tasks like visual querying [48] and potentially controlling smartphone functions [168, 125]. These models, trained on extensive datasets encompassing text, images, audio, and video, produces output that pattern matches in ways that meet user expectations, thus simulating reasoned responses. Just as a grocery list helps extend our working memory, a context-aware spatial agent could enhance our memory, cognition, and perception. Nonetheless, the technical limitations of latency and power remain significant challenges for real-time, in-context interaction.

Finally, integrating implicit input can help to make in-context computing better aware of user's attention. A number of mobile phone experiences and features seek to maximize user's engagement; even the mere presence of one's phone in the room can be distracting [142, 165]. The next phase in spatial computing offers a chance to redefine how our devices command our attention. By understanding user context and activity, future spatial interfaces could intelligently determine when to send notifications or proactively streamline multi-step tasks towards non-intrusive microinteractions [116]. Developing intent-aligned interaction models could decrease the attentional burden of our devices and allow them to one day act as an "extension of our unconscious" [166]. Spatial computing technologies are quickly becoming increasingly prevalent in our daily lives through the commercialization of technologies like smartphone AR, industrial AR, spatial audio earbuds, smart home sensors and smart speakers. As designers and engineers develop these systems, we have a unique opportunity to create interfaces to help users to be more productive in their tasks and foster greater engagement with the real world.

BIBLIOGRAPHY

[1] Ahuja, K., Kong, A., Goel, M., and Harrison, C. Direction-of-voice (dov) estimation for intuitive speech interaction with smart devices ecosystems. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2020), UIST '20, Association for Computing Machinery, p. 1121–1131.

[2] Allen, J. B., and Berkley, D. A. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America 65*, 4 (1979), 943–950.

[3] Altium Ltd. Altium Designer. `https://www.altium.com/altium-designer`.

[4] Amazon. Echo (3rd gen). `https://www.amazon.com/all-new-Echo/dp/B07NFTVP7P`.

[5] Amento, B., Hill, W., and Terveen, L. The sound of one hand: A wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI EA '02, Association for Computing Machinery, p. 724–725.

[6] Apple Inc. Apple airpods. `https://www.apple.com/airpods/`.

[7] Apple Inc. Apple vision pro. `https://www.apple.com/apple-vision-pro/`, Apr 2024.

[8] Ashbrook, D., Baudisch, P., and White, S. 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* (2011), pp. 2043–2046.

[9] Ashbrook, D., Lyons, K., and Starner, T. An investigation into round touchscreen wristwatch interaction. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services* (2008), pp. 311–314.

[10] Ashbrook, D. L. *Enabling mobile microinteractions*. Georgia Institute of Technology, 2010.

[11] Audun. Wireless timer synchronization among nrf5 devices. `https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/wireless-timer-synchronization-among-nrf5-devices`, Jul 2016.

[12] Azuma, R. T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments 6*, 4 (8 1997), 355–385.

[13] Biocca, F., Tang, A., Lamas, D., Gregg, J. L., Brady, R., and Gai, P. How do users organize virtual tools around their body in immersive virtual and augmented environments ? : An exploratory study of egocentric spatial mapping of virtual tools in the mobile infosphere. Tech. rep., Media Interface and Network Design Labs, 2003.

[14] Bluetooth Audio Telephony and Automotive Working Group. Hands-free profile: Bluetooth profile specification. Tech. Rep. v1.8, Bluetooth SIG, Apr 2020.

[15] Bluetooth SIG. Bluetooth le audio faqs. `https://www.bluetooth.com/media/le-audio/le-audio-faqs`.

[16] Bluetooth SIG. Bluetooth core specification v5.0. Tech. rep., Bluetooth SIG, 2016.

[17] Boring, S., Jurmu, M., and Butz, A. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7* (2009), pp. 161–168.

[18] Brandstein, M. *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2001.

[19] Bränzel, A., Holz, C., Hoffmann, D., Schmidt, D., Knaust, M., Lühne, P., Meusel, R., Richter, S., and Baudisch, P. Gravityspace: tracking users and their poses in a smart room using a pressure-sensing floor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), pp. 725–734.

[20] Braun, V., and Clarke, V. Using thematic analysis in psychology. *Qualitative Research in Psychology 3*, 2 (1 2006), 77–101.

[21] Brooks, F. P. Grasping reality through Illusion-Interactive graphics serving science. Tech. rep., NORTH CAROLINA UNIV AT CHAPEL HILL DEPT OF COMPUTER SCIENCE, Jan. 1988.

[22] Bui, H. Setting up the timeslot api. `https://devzone.nordicsemi.com/nordic/short-range-guides/b/software-development-kit/posts/setting-up-the-timeslot-api`, Jul 2015.

[23] Card, S. K., Robertson, G. G., and Mackinlay, J. D. The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1991), CHI '91, Association for Computing Machinery, p. 181–186.

[24] Caudell, T., and Mizell, D. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences* (1 1992), IEEE, pp. 659–669.

[25] Chan, E., Seyed, T., Stuerzlinger, W., Yang, X.-D., and Maurer, F. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 3403–3414.

[26] Chan, L., Chen, Y. L., Hsieh, C. H., Liang, R. H., and Chen, B. Y. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. *UIST 2015 - Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (11 2015), 549–556.

[27] Chatterjee, I. TouchPoint: A Wrist-Worn, On-Body Touch Interaction Device. Undergraduate Thesis. *Harvard College* (2016).

[28] Chatterjee, I., Khvan, O., Pforte, T., Li, R., and Patel, S. Augmented Silkscreen: Designing AR Interactions for Debugging Printed Circuit Boards. In *DIS 2021 - Proceedings of the 2021 ACM Designing Interactive Systems Conference: Nowhere and Everywhere* (2021).

[29] Chatterjee, I., Kim, M., Jayaram, V., Gollakota, S., Kemelmacher, I., Patel, S., and Seitz, S. M. Clearbuds: Wireless binaural earbuds for learning-based speech enhancement. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services* (New York, NY, USA, 2022), MobiSys '22, Association for Computing Machinery, p. 384–396.

[30] Chatterjee, I., Pforte, T., Tng, A., Salemi Parizi, F., Chen, C., and Patel, S. Ardw: An augmented reality workbench for printed circuit board debugging. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2022), UIST '22, Association for Computing Machinery.

[31] Chatterjee, I., Xiao, R., and Harrison, C. Gaze+gesture: Expressive, precise and targeted free-space interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (2015), pp. 131–138.

[32] Chen, Z., Xiao, X., Yoshioka, T., Erdogan, H., Li, J., and Gong, Y. Multi-channel overlapped speech recognition with location guided speech extraction network. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (2018), IEEE, pp. 558–565.

[33] Chhetri, A., Hilmes, P., Kristjansson, T., Chu, W., Mansour, M., Li, X., and Zhang, X. Multichannel audio front-end for far-field automatic speech recognition. In *2018 EUSIPCO* (2018), IEEE, pp. 1527–1531.

[34] Cox, R. V., De Campos Neto, S. F., Lamblin, C., and Sherif, M. H. In *ITU-T coders for wideband, superwideband, and fullband speech communication [Series Editorial]* (2009), vol. 47, pp. 106–109.

[35] Defossez, A., Synnaeve, G., and Adi, Y. Real time speech enhancement in the waveform domain.

[36] Dey, A. K. Understanding and using context. *Personal and ubiquitous computing 5* (2001), 4–7.

[37] Ding, Y., Shultz, C., and Harrison, C. Surface i/o: Creating devices with functional surface geometry for haptics and user input. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–22.

[38] Drew, D., Newcomb, J. L., McGrath, W., Maksimovic, F., Mellis, D., and Hartmann, B. The Toastboard. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (New York, NY, USA, 10 2016), ACM, pp. 677–686.

[39] Duan, Z., Mysore, G., and Smaragdis, P. Speech enhancement by online non-negative spectrogram decomposition in non-stationary noise environments. In *13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012* (Dec. 2012), 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012, pp. 594–597. 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012 ; Conference date: 09-09-2012 Through 13-09-2012.

[40] Feiner, S., Macintyre, B., and Seligmann, D. Knowledge-based augmented reality. *Communications of the ACM 36*, 7 (7 1993), 53–62.

[41] Frost, O. L. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE 60*, 8 (1972), 926–935.

[42] Fu, S.-W., Liao, C.-F., Tsao, Y., and Lin, S.-D. Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement.

[43] Fukumoto, M., and Suenaga, Y. "FingeRing": A full-time wearable interface. *Conference on Human Factors in Computing Systems - Proceedings 1994-April* (4 1994), 81–82.

[44] Germain, F. G., Chen, Q., and Koltun, V. Speech denoising with deep feature losses.

[45] Gong, J., Gupta, A., and Benko, H. Acustico: surface tap detection and localization using wrist-based acoustic tdoa sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (2020), pp. 406–419.

[46] Gong, J., Zhang, Y., Zhou, X., and Yang, X.-D. Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (2017).

[47] Google. Google meet. `https://meet.google.com`.

[48] Google. Google lens - search what you see. `https://lens.google/`, Jan 2024.

[49] Goyal, P., Agrawal, H., Paradiso, J. A., and Maes, P. BoardLab. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology - UIST '13 Adjunct* (New York, New York, USA, 2013), ACM Press, pp. 19–20.

[50] Greenberg, S., and Buxton, B. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, Association for Computing Machinery, p. 111–120.

[51] Greenwold, S. Spatial computing. masters thesis. *Massachusetts Institute of Technology* (2003).

[52] Gu, R., Zhang, S.-X., Chen, L., Xu, Y., Yu, M., Su, D., Zou, Y., and Yu, D. Enhancing end-to-end multi-channel speech separation via spatial feature learning. *arXiv preprint arXiv:2003.03927* (2020).

[53] Gu, Y., Yu, C., Li, Z., Li, W., Xu, S., Wei, X., and Shi, Y. Accurate and low-latency sensing of touch contact on any surface with finger-worn imu sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 10 2019), UIST '19, Association for Computing Machinery, p. 1059–1070.

[54] Hahn, J., Ludwig, B., and Wolff, C. Augmented reality-based training of the PCB assembly process. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia* (New York, NY, USA, 11 2015), vol. 30-Novembe, ACM, pp. 395–399.

[55] Han, C., Luo, Y., and Mesgarani, N. Real-time binaural speech separation with preserved spatial cues.

[56] Harrison, C. *The Human Body as an Interactive Computing Platform*. PhD thesis, Carnegie Mellon University, USA, 2013. AAI3578670.

[57] Harrison, C., and Hudson, S. E. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), pp. 205–208.

[58] Hirsh, I. J. The Influence of Interaural Phase on Interaural Summation and Inhibition. *The Journal of the Acoustical Society of America 20*, 4 (06 2005), 536–544.

[59] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications.

[60] Huang, D.-Y., Chan, L., Yang, S., Wang, F., Liang, R.-H., Yang, D.-N., Hung, Y.-P., and Chen, B.-Y. 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* (2016), pp. 1526–1537.

[61] InspectAR. inspectAR Augmented Reality PCB Tools. `https://www.inspectar.com/`.

[62] International Telecommunication Union. Series G: Transmission Systems and Media, Digital Systems and Networks. Tech. rep., Telecommunication Standardization Sector of ITU, 2003.

[63] InvenSense. Microphone array beamforming. Tech. Rep. AN-1140-00, InvenSense Inc., 1745 Technology Drive, San Jose, CA 95110 U.S.A, December 2013.

[64] Iravantchi, Y., Goel, M., and Harrison, C. BeamBand: Hand gesture sensing with ultrasonic beamforming. *Conference on Human Factors in Computing Systems - Proceedings* (5 2019).

[65] Iravantchi, Y., Zhao, Y., Kin, K., and Sample, A. P. Sawsense: Using surface acoustic waves for surface-bound event recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–18.

[66] Jenrungrot, T., Jayaram, V., Seitz, S., and Kemelmacher-Shlizerman, I. The cone of silence: Speech separation by localization.

[67] Jiang, X., Zhu, L., Liu, J., and Song, A. A slam-based 6dof controller with smooth auto-calibration for virtual reality. *The Visual Computer* (2022), 1–14.

[68] Karchemsky, M., Zamfirescu-Pereira, J. D., Wu, K.-J., Guimbretière, F., and Hartmann, B. Heimdall: A Remotely Controlled Inspection Workbench For Debugging Microcontroller Projects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–12.

[69] Kawsar, F., Min, C., Mathur, A., and Montanari, A. Earables for personal-scale behavior analytics. *IEEE Pervasive Computing 17*, 3 (2018), 83–89.

[70] Khurana, R., and Hodges, S. Beyond the Prototype: Understanding the Challenge of Scaling Hardware Device Production. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 4 2020), ACM, pp. 1–11.

[71] KiCad Team. KiCad EDA - Schematic Capture & PCB Design Software. `https://kicad-pcb.org/`.

[72] Kienzle, W., and Hinckley, K. Lightring: always-available 2d input on any surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (2014), pp. 157–160.

[73] Kienzle, W., Whitmire, E., Rittaler, C., and Benko, H. Electroring: Subtle pinch and touch detection with a ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–12.

[74] Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. 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* (New York, NY, USA, 2012), UIST '12, Association for Computing Machinery, p. 167–176.

[75] Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. 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* (New York, NY, USA, 2012), UIST '12, Association for Computing Machinery, p. 167–176.

[76] Kim, J., Kim, M., Lee, W. S., and Yoon, S. H. Vibaware: Context-aware tap and swipe gestures using bio-acoustic sensing. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (2023), IEEE, pp. 609–610.

[77] Kim, Y., Choi, Y., Lee, H., Lee, G., and Bianchi, A. VirtualComponent. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 5 2019), ACM, pp. 1–13.

[78] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[79] Kirsh, D. The intelligent use of space. *Artif. Intell. 73*, 1-2 (Feb. 1995), 31–68.

[80] Klein, G., and Murray, D. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007), pp. 225–234.

[81] Kock, W. Binaural localization and masking. *The Journal of the Acoustical Society of America 22*, 6 (1950), 801–804.

[82] Kress, B. C., and Chatterjee, I. Waveguide combiners for mixed reality headsets: a nanophotonics design perspective. *Nanophotonics 10*, 1 (2020), 41–74.

[83] Krim, H., and Viberg, M. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine 13*, 4 (1996), 67–94.

[84] Krisp. Krisp ai - your ai-powered assistant for meetings and calls. `https://www.krisp.ai`.

[85] Kuno, W., Sugiura, Y., Asano, N., Kawai, W., and Sugimoto, M. 3d reconstruction of hand postures by measuring skin deformation on back hand. In *Proceedings of the 27th International Conference on Artificial Reality and Telexistence and 22nd Eurographics Symposium on Virtual Environments* (Goslar, DEU, 2017), ICAT-EGVE '17, Eurographics Association, p. 221–228.

[86] Kuo, L.-C., Chiu, H.-Y., Chang, C.-W., Hsu, H.-Y., and Sun, Y.-N. Functional workspace for precision manipulation between thumb and fingers in normal hands. *Journal of electromyography and kinesiology 19*, 5 (2009), 829–839.

[87] Landis, D., and Ivey, L. Npr & edison research: Smart speaker ownership reaches 35 *NPR* (Jun 2022).

[88] Lewis, C. Using the "thinking Aloud" Method in Cognitive Interface Design. *IBM Research Report, RC-9265 9265* (1982).

[89] Li, J., Sakamoto, S., Hongo, S., Akagi, M., and Suzuki, Y. Two-stage binaural speech enhancement with wiener filter for high-quality speech communication. *Speech Communication 53*, 5 (2011), 677–689.

[90] Li, Z.-M., and Tang, J. Coordination of thumb joints during opposition. *Journal of biomechanics 40*, 3 (2007), 502–510.

[91] Liang, C., Yu, C., Qin, Y., Wang, Y., and Shi, Y. DualRing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5*, 3 (9 2021), 27.

[92] Lin, J. W., Wang, C., Huang, Y. Y., Chou, K. T., Chen, H. Y., Tseng, W. L., and Chen, M. Y. BackHand: Sensing hand gestures via back of the hand. *UIST 2015 - Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (11 2015), 557–564.

[93] Lin, R., Ramesh, R., Iannopollo, A., Vincentelli, A. S., Dutta, P., Alon, E., and Hartmann, B. Beyond Schematic Capture Meaningful Abstractions for Better Electronics Design Tools. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019).

[94] Loclair, C., Gustafson, S., and Baudisch, P. Pinchwatch: a wearable device for one-handed microinteractions. In *Proc. MobileHCI* (2010), vol. 10, Citeseer.

[95] Luo, Y., Chen, Z., Mesgarani, N., and Yoshioka, T. End-to-end microphone permutation and number invariant multi-channel speech separation, 2020.

[96] Luo, Y., and Mesgarani, N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2019).

[97] Lyon, R. A computational model of binaural localization and separation. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing* (1983), vol. 8, IEEE, pp. 1148–1151.

[98] Macartney, C., and Weyde, T. Improved speech enhancement with the wave-u-net.

[99] Macpherson, E. A., and Middlebrooks, J. C. Listener weighting of cues for lateral angle: the duplex theory of sound localization revisited. *The Journal of the Acoustical Society of America 111*, 5 (2002), 2219–2236.

[100] Magic Leap. Magic leap - device. `https://www.magicleap.com/magic-leap-2`.

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

[102] Meier, M., Streli, P., Fender, A., and Holz, C. TaplD: Rapid touch interaction in virtual reality using wearable sensing. *Proceedings - 2021 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2021* (3 2021), 519–528.

[103] Mercer, D. Internet of Things Now Numbers 22 Billion Devices But Where Is The Revenue? *Business Wire* (2019), Newsroom>Press Releases.

[104] Meta Platforms, Inc. Meta Quest 2: Our Most Advanced New All-in-One VR Headset Meta Store. `https://store.facebook.com/quest/products/quest-2/`, 2022.

[105] Microsoft. Deep Noise Suppression Challenge – Interspeech 2021. `https://www.microsoft.com/en-us/research/academic-program/deep-noise-suppression-challenge-interspeech-2021/`, 2021.

[106] Microsoft. Hand tracking - mrtk 2, 2022.

[107] Milgram, P., and Kishino, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems 77*, 12 (1994), 1321–1329.

[108] Miller, R. B. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I* (New York, NY, USA, 1968), AFIPS '68 (Fall, part I), Association for Computing Machinery, p. 267–277.

[109] Mohammadiha, N., Smaragdis, P., and Leijon, A. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *IEEE Transactions on Audio, Speech, and Language Processing 21*, 10 (Oct 2013), 2140–2151.

[110] Muensterer, O. J., Lacher, M., Zoeller, C., Bronstein, M., and Kübler, J. Google Glass in pediatric surgery: An exploratory study. *International Journal of Surgery 12*, 4 (4 2014), 281–289.

[111] Mujibiya, A., Cao, X., Tan, D. S., Morris, D., Patel, S. N., and Rekimoto, J. The Sound of Touch: On-body Touch and Gesture Sensing Based on Transdermal Ultrasound Propagation. *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces* (2013).

[112] Nguyen, A., and Banic, A. 3dtouch: A wearable 3d input device for 3d applications. In *2015 IEEE Virtual Reality (VR)* (2015), pp. 373–373.

[113] Nikzad, M., Nicolson, A., Gao, Y., Zhou, J., Paliwal, K. K., and Shang, F. Deep residual-dense lattice network for speech enhancement.

[114] Ochiai, Y. The visible electricity device: visible breadboard. In *ACM SIGGRAPH 2010 Posters* (New York, NY, USA, 2010), SIGGRAPH '10, Association for Computing Machinery.

[115] Oh, S. Y., Yoon, B., and Woo, W. Finger Contact in Gesture Interaction Improves Time-domain Input Accuracy in HMD-based Augmented Reality. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), 1–8.

[116] Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), pp. 919–928.

[117] Paine, T. L., Khorrami, P., Chang, S., Zhang, Y., Ramachandran, P., Hasegawa-Johnson, M. A., and Huang, T. S. Fast wavenet generation algorithm.

[118] Pascual, S., Bonafonte, A., and Serrà, J. Segan: Speech enhancement generative adversarial network.

[119] Pasunuri, A., Jessurun, N., Dizon-Paradis, O. P., and Asadizanjani, N. A comparison of neural networks for pcb component segmentation. In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2021), pp. 113–123.

[120] Peterson, M. Apple airpods, beats dominated audio wearable market in 2020. `https://appleinsider.com/articles/21/03/30/apple-airpods-beats-dominated-audio-wearable-market-in-2020`, 2021.

[121] Piaget, J., and Cook, M. t. *The construction of reality in the child*, vol. 386. Basic Books, New York, NY, US, 1954.

[122] Pollack, I., and Pickett, J. M. Cocktail party effect. *The Journal of the Acoustical Society of America 29*, 11 (1957), 1262–1262.

[123] Project Gutenberg. Project gutenberg. `https://www.gutenberg.org/`. Accessed: 2021-12-20.

[124] qu1ck. GitHub - openscopeproject/InteractiveHtmlBom: Interactive HTML BOM generation plugin for KiCad. `https://github.com/openscopeproject/InteractiveHtmlBom`.

[125] rabbit research team. Learning human actions on computer applications, 2023.

[126] Reddy, C. K. A., Dubey, H., Gopal, V., Cutler, R., Braun, S., Gamper, H., Aichner, R., and Srinivasan, S. Icassp 2021 deep noise suppression challenge. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 6623–6627.

[127] Reindl, K., Zheng, Y., and Kellermann, W. Speech enhancement for binaural hearing aids based on blind source separation. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)* (2010), IEEE, pp. 1–6.

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

[129] Rico, J., and Brewster, S. Usable gestures for mobile interfaces: evaluating social acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), pp. 887–896.

[130] Risoud, M., Hanson, J.-N., Gauvrit, F., Renard, C., Lemesre, P.-E., Bonne, N.-X., and Vincent, C. Sound source localization. *European Annals of Otorhinolaryngology, Head and Neck Diseases 135*, 4 (2018), 259–264.

[131] Robotas Technologies Ltd. Mascot | Robotas.

[132] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation.

[133] Rosenthal, S., Kane, S. K., Wobbrock, J. O., and Avrahami, D. Augmenting on-screen instructions with micro-projected guides: when it works, and when it fails. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (2010), pp. 203–212.

[134] Roux, J. L., Wisdom, S., Erdogan, H., and Hershey, J. R. SDR - half-baked or well done? *CoRR abs/1811.02508* (2018).

[135] Samsung. Galaxy s5 explained: Audio. https://news.samsung.com/global/galaxy-s5-explained-audio, Jun 2014.

[136] Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (2009), pp. 167–176.

[137] Scheibler, R., Bezzam, E., and Dokmanić, I. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 351–355.

[138] Schott, G. D. Penfield's homunculus: a note on cerebral cartography. *Journal of neurology, neurosurgery, and psychiatry 56*, 4 (1993), 329.

[139] Schwerdtfeger, B., and Klinker, G. Supporting order picking with Augmented Reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (9 2008), IEEE, pp. 91–94.

[140] Sennheiser. Earbuds that put sound first. `https://en-de.sennheiser.com/newsroom/earbuds-that-put-sound-first`, Mar 2020.

[141] Shi, Y., Zhang, H., Zhao, K., Cao, J., Sun, M., and Nanayakkara, S. Ready, steady, touch! sensing physical contact with a finger-mounted imu. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4*, 2 (2020), 1–25.

[142] Skowronek, J., Seifert, A., and Lindberg, S. The mere presence of a smartphone reduces basal attentional performance. *Scientific Reports 13*, 1 (2023), 9363.

[143] Soni, M. H., Shah, N., and Patil, H. A. Time-frequency masking-based speech enhancement using generative adversarial network. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 5039–5043.

[144] Soukoreff, R. W., and MacKenzie, I. S. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *Int. J. Hum.-Comput. Stud. 61*, 6 (dec 2004), 751–789.

[145] Strasnick, E., Agrawala, M., and Follmer, S. Scanalog: Interactive Design and Debugging of Analog Circuits with Programmable Hardware. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 10 2017), ACM, pp. 321–330.

[146] Strasnick, E., Follmer, S., and Agrawala, M. Pinpoint. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 5 2019), ACM, pp. 1–11.

[147] Stöter, F.-R., Liutkus, A., and Ito, N. The 2018 signal separation evaluation campaign.

[148] Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., and Zhong, J. Attention is all you need in speech separation, 2021.

[149] Sun, X., Xia, R., Li, J., and Yan, Y. A deep learning based binaural speech enhancement approach with spatial cues preservation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 5766–5770.

[150] Sutherland, I. E. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I* (New York, NY, USA, 1968), AFIPS '68 (Fall, part I), Association for Computing Machinery, p. 757–764.

[151] Takada, R., Shizuki, B., and Kadomoto, J. A sensing technique for data glove using conductive fiber. *Conference on Human Factors in Computing Systems - Proceedings* (5 2019).

[152] Tan, K., Zhang, X., and Wang, D. Real-time speech enhancement using an efficient convolutional recurrent network for dual-microphone mobile phones in close-talk scenarios. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 5751–5755.

[153] Tan, K., Zhang, X., and Wang, D. Deep learning based real-time speech enhancement for dual-microphone mobile phones. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2021), 1–1.

[154] Tang, A., Owen, C., Biocca, F., and Mou, W. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the conference on Human factors in computing systems - CHI '03* (New York, New York, USA, 2003), ACM Press, p. 73.

[155] Tzirakis, P., Kumar, A., and Donley, J. Multi-channel speech enhancement using graph neural networks.

[156] UltraLeap. Tracking Leap Motion Controller. https://www.ultraleap.com/product/leap-motion-controller/, 2021.

[157] van Hoesel, R., Böhm, M., Pesch, J., Vandali, A., Battmer, R. D., and Lenarz, T. Binaural speech unmasking and localization in noise with bilateral cochlear implants using envelope and fine-timing based strategies. *The Journal of the Acoustical Society of America 123*, 4 (2008), 2249–2263.

[158] Van Veen, B. D., and Buckley, K. M. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine 5*, 2 (1988), 4–24.

[159] Varjo Technologies. Varjo XR-3 - the industry's highest resolution mixed reality headset. https://varjo.com/products/xr-3/, Nov. 2020.

[160] Veaux, C., Yamagishi, J., MacDonald, K., et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

[161] Vertelney, L. Using video to prototype user interfaces. *ACM SIGCHI Bulletin 21*, 2 (10 1989), 57–61.

[162] Waghmare, A., Ben Taleb, Y., Chatterjee, I., Narendra, A., and Patel, S. Z-ring: Single-point bio-impedance sensing for gesture, touch, object and user recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–18.

[163] Waghmare, A., Boldu, R., Whitmire, E., and Kienzle, W. Optiring: Low-resolution optical sensing for subtle thumb-to-index micro-interactions. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction* (2023), pp. 1–13.

[164] Wang, D. On ideal binary mask as the computational goal of auditory scene analysis. In *Speech separation by humans and machines*. Springer, 2005, pp. 181–197.

[165] Ward, A. F., Duke, K., Gneezy, A., and Bos, M. W. Brain drain: The mere presence of one's own smartphone reduces available cognitive capacity. *Journal of the association for consumer research 2*, 2 (2017), 140–154.

[166] Weiser, M. Computer science challenges for the next 10 years, Nov. 1996.

[167] Wellner, P. Interacting with paper on the DigitalDesk. *Communications of the ACM 36*, 7 (7 1993), 87–96.

[168] Wen, H., Li, Y., Liu, G., Zhao, S., Yu, T., Li, T. J.-J., Jiang, S., Liu, Y., Zhang, Y., and Liu, Y. Autodroid: Llm-powered task automation in android, 2024.

[169] Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Roux, J., Hershey, J. R., and Schuller, B. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation - Volume 9237* (Berlin, Heidelberg, 2015), LVA/ICA 2015, Springer-Verlag, p. 91–99.

[170] Westhausen, N. L., and Meyer, B. T. Dual-signal transformation lstm network for real-time noise suppression, arxiv, 2020.

[171] Whitmire, E. pyrealtime/filter_layers.py at master · ewhitmire/pyrealtime · GitHub. `https://github.com/ewhitmire/pyrealtime/blob/master/pyrealtime/filter_layers.py`.

[172] Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Roux, J. L. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160* (2019).

[173] Wobbrock, J. O., Shinohara, K., and Jansen, A. The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, Association for Computing Machinery, p. 1639–1648.

[174] Wolf, K., Naumann, A., Rohs, M., and Müller, J. A taxonomy of microinteractions: Defining microgestures based on ergonomic and scenario-dependent requirements. No. Part I in 13th International Conference on Human-Computer Interaction (INTERACT), Springer, pp. 559–575.

[175] Wu, P.-C., Wang, R., Kin, K., Twigg, C., Han, S., Yang, M.-H., and Chien, S.-Y. DodecaPen: Accurate 6DoF Tracking of a Passive Stylus. *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (10 2017), 365–374.

[176] Wu, T.-Y., Shen, H.-P., Wu, Y.-C., Chen, Y.-A., Ku, P.-S., Hsu, M.-W., Liu, J.-Y., Lin, Y.-C., and Chen, M. Y. CurrentViz. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 10 2017), ACM, pp. 343–349.

[177] Xu, C., Zhou, B., Krishnan, G., and Nayar, S. Ao-finger: Hands-free fine-grained finger gesture recognition via acoustic-optic sensor fusing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–14.

[178] Xu, Y., Du, J., Dai, L.-R., and Lee, C.-H. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 23*, 1 (2015), 7–19.

[179] Yang, X.-D., Grossman, T., Wigdor, D., and Fitzmaurice, G. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), pp. 147–156.

[180] Yoshioka, T., Erdogan, H., Chen, Z., and Alleva, F. Multi-microphone neural speech separation for far-field multi-talker speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5739–5743.

[181] Zhai, S., Milgram, P., and Buxton, W. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1996), pp. 308–315.

[182] Zhang, C., Waghmare, A., Kundra, P., Pu, Y., Gilliland, S., Ploetz, T., Starner, T. E., Inan, O. T., and Abowd, G. D. Fingersound: Recognizing unistroke thumb gestures using a ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1*, 3 (sep 2017).

[183] Zhang, C., Xue, Q., Waghmare, A., Jain, S., Pu, Y., Hersek, S., Lyons, K., Cunefare, K. A., Inan, O. T., and Abowd, G. D. SoundTrak. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1*, 2 (6 2017), 1–25.

[184] Zhang, C., Xue, Q., Waghmare, A., Meng, R., Jain, S., Han, Y., Li, X., Cunefare, K., Ploetz, T., Starner, T., et al. Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–10.

[185] Zhang, C., Xue, Q., Waghmare, A., Meng, R., Jain, S., Han, Y., Li, X., Cunefare, K., Ploetz, T., Starner, T., Inan, O., and Abowd, G. D. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018).

[186] Zhang, T., Zeng, X., Zhang, Y., Sun, K., Wang, Y., and Chen, Y. ThermalRing: Gesture and Tag Inputs Enabled by a Thermal Imaging Smart Ring. *Conference on Human Factors in Computing Systems - Proceedings* (4 2020).

[187] Zhang, X., and Wang, D. Deep learning based binaural speech separation in reverberant environments. *IEEE/ACM transactions on audio, speech, and language processing 25*, 5 (2017), 1075–1084.

[188] Zhang, Y., Yang, C., Hudson, S. E., Harrison, C., and Sample, A. Wall++ room-scale interactive and context-aware sensing. In *Proceedings of the 2018 chi conference on human factors in computing systems* (2018), pp. 1–15.

[189] Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., and Torralba, A. The sound of pixels.

[190] Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A hand gesture interface device. *ACM Sigchi Bulletin 18*, 4 (1986), 189–192.