BLEselect: Gestural IoT Device Selection via Bluetooth Angle of Arrival Estimation from Smart Glasses

TENGXIANG ZHANG*, Institute of Computing Technology, CAS and UCAS, China
ZITONG LAN†, Southeast University, China
CHENREN XU, Peking University, China
YANRONG LI and YIQIANG CHEN*, Institute of Computing Technology, CAS and UCAS, China

Spontaneous selection of IoT devices from the head-mounted device is key for user-centered pervasive interaction. *BLEselect* enables users to select an unmodified Bluetooth 5.1 compatible IoT device by nodding at, pointing at, or drawing a circle in the air around it. We designed a compact antenna array that fits on a pair of smart glasses to estimate the Angle of Arrival (AoA) of IoT and wrist-worn devices' advertising signals. We then developed a sensing pipeline that supports all three selection gestures with lightweight machine learning models, which are trained in real-time for both hand gestures. Extensive characterizations and evaluations show that our system is accurate, natural, low-power, and privacy-preserving. Despite the small effective size of the antenna array, our system achieves a higher than 90% selection accuracy within a 3 meters distance in front of the user. In a user study that mimics real-life usage cases, the overall selection accuracy is 96.7% for a diverse set of 22 participants in terms of age, technology savviness, and body structures.

CCS Concepts: • **Human-centered computing → Interaction devices**; **Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: IoT, device selection, gesture, angle of arrival

ACM Reference Format:
Tengxiang Zhang, Zitong Lan, Chenren Xu, Yanrong Li, and Yiqiang Chen. 2022. BLEselect: Gestural IoT Device Selection via Bluetooth Angle of Arrival Estimation from Smart Glasses. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 198 (December 2022), 28 pages. https://doi.org/10.1145/3569482

1 INTRODUCTION

Recent research shows that the combination of the Internet of Things (IoT) and Head-mounted Devices (HMD) greatly improves interaction experience and working efficiency in homes, factories, storehouses, and shops [15, 18, 22, 33]. Ubiquitously deployed IoT devices sense and actuate the physical environment, while associated digital information is transmitted to an HMD like a pair of smart glasses for spontaneous display. However, simultaneous display of all information from numerous IoT devices leads to information overload. Natural and efficient IoT device selection can filter the overwhelming information. Users can select one item to display its information in the HMD's limited Field of View (FOV). Example scenarios include a user selects one appliance to

*Tengxiang Zhang, and Yiqiang Chen are also with Beijing Key Lab. of Mobile Computing and Pervasive Device, and Shandong Institute of Industrial Technology. Correspondence to: Tengxiang Zhang and Yiqiang Chen.
†Zitong Lan was a research intern at Institute of Computing Technology, Chinese Academy of Sciences during the period of the project.

Authors' addresses: Tengxiang Zhang, zhangtengxiang@ict.ac.cn, Institute of Computing Technology, CAS and UCAS, Beijing, China; Zitong Lan, zitonglan1@gmail.com, Southeast University, Nanjing, China; Chenren Xu, Peking University, Beijing, China, chenren.xu@gmail.com; Yanrong Li, kidominox@gmail.com; Yiqiang Chen, chenyiqiang@gmail.com, Institute of Computing Technology, CAS and UCAS, Beijing, China.

2474-9567/2022/12-ART198
https://doi.org/10.1145/3569482

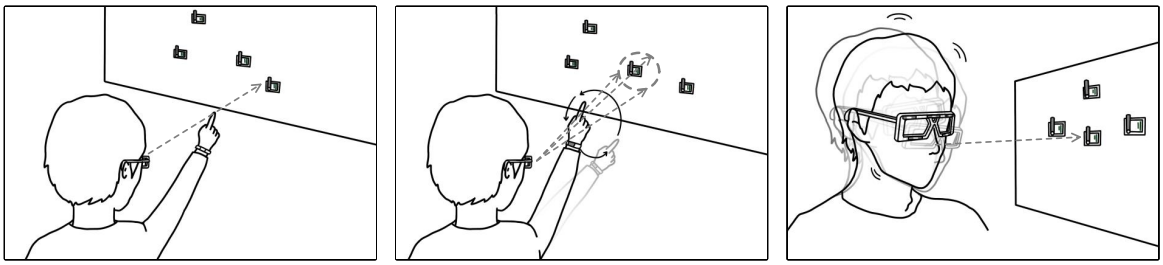

Fig. 1. BLEselect supports IoT device selection using three types of gestures: *Pointing*, *Encircling* and *Nodding*

display its functions in a smart home, selects one commodity to display its detailed information in a shop, or selects one machine component to display its current status in a factory.

Researchers have proposed various HMD-based IoT device selection methods. Most HMDs are equipped with microphones so that users can issue voice commands for selection. However, voice commands are inappropriate in quiet environments, vulnerable to surrounding noises, and inefficient to convey locations of the target device. GUI-based selection methods display all surrounding IoT devices as icons or a list in the HMD display. This will occupy a large portion of the already limited FOV of HMDs though, which causes severe visual occlusions. Gestural IoT device selection in the physical space, on the other hand, enables users to select IoT devices by looking at or pointing at the target device. However, existing HMD-based selection techniques either require extra hardware (*e.g.*, IR transceivers [48], speakers [42]) or rely on environmental infrastructures(*e.g.*, WiFi router [24], RFID reader [22]). Many of them [20, 22, 24, 32] use video signals to recognize objects or barcodes, and detect gestures. Continuous video recording also raises severe privacy concerns especially from HMDs. The recording and image processing also consumes lots of power, which can drain the HMD's battery quickly. It is thus necessary to develop an infrastructure-free HMD-based IoT device selection technique that enables users to accurately select the IoT device using familiar gestures without privacy concerns, while requiring minimal additional hardware components and power consumption on both the HMD and IoT devices.

So we propose *BLEselect*, a novel gestural IoT device selection method that is natural, accurate, privacy-preserving, and infrastructure-free. BLEselect fits a carefully designed conformal antenna array onto the frames of a pair of smart glasses, which measures the Angles of Arrival (AoA) of advertising signals from Bluetooth 5.1 compatible IoT devices [1]. The system detects the head *Nodding* gesture and its direction by analyzing AoA changes of surrounding IoT devices so that users can nod to select the device just as they nod at a person. When combined with a wrist-worn device (*e.g.*, a commercial smartwatch or a smart ring), our system can monitor finger pointing directions and track hand movement trajectories. Users can then select the target device by *Pointing* at or drawing a circle in the air around it (*i.e.*, *Encircling*, Figure 1). Through a carefully designed sensing pipeline, *BLEselect* supports all three **natural** selection gestures so that users can choose the gesture that best suits the context. Despite the AoA estimation errors due to the inherent Bluetooth phase measurement errors [12] and the small antenna array aperture, our system still achieves **accurate** device selection by using one-class SVM models to map the head/hand orientation with the target device. Our system is also **privacy-preserving** and **infrastructure-free** since it solely relies on pair-to-pair Bluetooth communications, which eliminates privacy concerns and configuration troubles. BLEselect achieves all the above desirable features while requiring minimal resources on either the IoT device or the HMD. It works with existing Bluetooth 5.1 compatible IoT devices without any hardware modification. On the HMD side, it only adds an RF switch and several antenna elements.

[1] https://www.bluetooth.com/bluetooth-resources/bluetooth-direction-finding/

Thanks to the low power nature of the Bluetooth technology, BLEselect adds low power consumption to both the HMD and the IoT devices.

It is challenging to estimate AoA of IoT devices from a pair of smart glasses due to the size constrain. We conducted simulations to design a compact 5-element 2.4GHz antenna array that fits on the frame of a pair of smart glasses (145mm in length and 60mm in height) without occluding the eyesight. Despite the small effective size of the antenna array, our system has AoA estimation errors of 2.8°-4.9°, 4.0°-6.7°, 6.0°-8.8°, and 9.6°-13.1°at 1m, 2m, 3m, and 5m distances respectively, which increases from the center to corners of the FOV($|\theta_{azimuth}| \leq 30°, |\phi_{elevation}| \leq 20°$). Another technical challenge is to ensure robust selection performance against hardware variances and environmental differences. We leverage the wrist-worn device to offset system errors and signal variations caused by environmental changes. More specifically, we train a new lightweight one-class SVM classifier with AoA data only from the just performed hand selection gesture. The model takes the AoA data of each received IoT device as input and outputs whether it is the device user intends to select. Our selection experiments in three indoor environments show that the average selection accuracy is higher than 90% within a 3m distance between the receiver and the transmitter for all three gestures. In the user study with more realistic settings, *Encircling*, *Pointing*, and *Nodding* has an selection accuracy of 97.8%, 97.4%, and 94.8% respectively for a diverse set of 22 participants with vastly different ages, technology savviness, and body heights. The average power consumption of an IoT transmitter advertising at 25Hz is 1.9mW, and that of the receiver on the HMD is only 9.3mW when tracking seven transmitters, which is far less than the typical power consumption of an HMD [23] [2]. We summarize our major contributions in the design and implementation of BLEselect as follows:

(1) Methodology-wise, we proposed a natural, accurate, and privacy-preserving HMD-based IoT device selection method, which novelly leverages the AoA of wireless signals of IoT devices. It works with unmodified Bluetooth 5.1 devices and adds minimal hardware and power consumption to the HMD.
(2) Technical-wise, we designed a compact 5-element 2.4GHz antenna array that fits on the frame of smart glasses through simulation; we also developed a novel device selection method that trains machine learning models in real-time with AoA data generated from the hand gesture, which offsets systems errors and outperforms the conventional angular distance-based method.
(3) System-wise, we implemented a complete prototype system that supports device selection using the three natural gestures; we also conducted extensive experiments and user studies, which validated that our system's performance is accurate, robust, and low-power in various environments and location settings.

2 BACKGROUND AND RELATED WORK

In this section, we first briefly introduce the Bluetooth protocol and its direction finding principle. We then explain why we choose Bluetooth by comparing it with the direction finding techniques using other wireless signals. At last, we review existing IoT device selection methods and show BLEselect's unique position in the relevant literature.

2.1 Bluetooth Direction Finding Primer

Bluetooth is a widely used wireless communication protocol by IoT devices for its low cost (3-5 USD), low power (10-20mW when TX/RX is active), and flexible connection/connectionless communication mechanisms. It is estimated that 14.6 billion Bluetooth IoT devices will be in use worldwide in 2022, which amounts to 35% of all personal IoT devices [3]. The recent Bluetooth 5.1 protocol supports Angle of Arrival (AoA) measurements during connection-less communications, which our system leverages for IoT device selection.

[2]Google Glass has a power consumption of 332mW with the system active and the screen off [23].
[3]https://www.bluetooth.com/17c11db333068c959a20568f2ff92e2c/

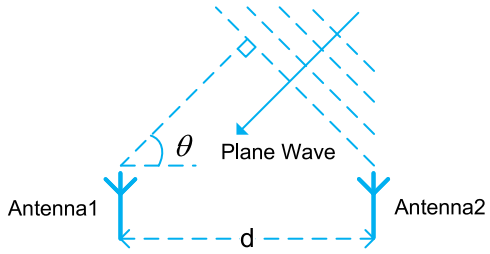

Fig. 2. Schematic diagram of AoA mechanism

Under a connection-less AoA setting, Bluetooth transmitters on IoT devices will advertise periodically while the Bluetooth receiver on the HMD will scan for such advertisements. The receiver synchronizes with the periodic advertisements of multiple IoT devices through time division. Once synced, the receiver switches its single radio frond-end between different antenna elements in an antenna array. The system calculates the phases of the Continuous Tone Extension (CTE) part of the advertising packet based on the IQ value reported at each antenna element and interpolates the data to estimate the phases at each antenna element over the entire CTE signal [30, 47].

The basic AoA estimation mechanism assumes that the radio signal is a single-tone plane wave impinging on the receiving array. As Figure 2 shows, the phase difference ϕ between two adjacent antennas is $\phi = 2\pi d \cos\theta/\lambda$, where λ is the wavelength, d the distance between antennas. The AoA of incoming signal θ can be calculated with the following geometric equation,

$$\theta = \arccos(\frac{\lambda\phi}{2\pi d}). \tag{1}$$

2.2 Existing Direction Finding Techniques

Researchers have used various signals for AoA estimation and localization purposes. RFID works at the 900MHz band and is widely used for inventory applications like commodity localization [16, 39, 40]. However, the RFID reader usually consumes large amounts of power (several Watts) to activate the tags and compensate for the weak back-scattered signals. SociTrack [8] uses Bluetooth for communication and Ultra Wide Band (UWB) radio for precise localization to reduce reader power consumption while also achieving a high localization accuracy. RF-echo [11] uses OFDM active reflection to improve the signal-to-noise ratio of the backs-cattered signals. Previous work also leverages existing infrastructures like WiFi for localization purposes. ArrayTrack [44] leverages MIMO radios and combines information from multiple Access Points (AP) for mobile device localization. SpotFi [21] achieves decimeter level localization accuracy with an AP that only has three antennas. TagFi [35] uses a WIFi AP and a laptop to localize OOK modulated ultra-low-power tags. VisIoT [19] uses two antennas and a gyroscope to calculate a Zigbee transceiver's azimuth and elevation angles and maps it onto the 2D camera image. Millimetro [36] achieves centimeter-level localization accuracy at long ranges by leveraging the high bandwidth of mmWave signals. Ashok et al. [5] uses a radio-optical hybrid system that achieves both low-power and accurate beacon AoA estimation. However, it does not work with existing IoT devices. Also, the tags need to be carefully placed to provide line-of-sights for successful localization. Other than RF signals, researchers have also used audio [7, 41] and Passive Infrared (PIR) signals [25] for AoA estimation. However, direction finding in the UHF RFID or the millimeter bands requires the deployment of power-hungry readers. Speakers/microphones and PIR sensors, on the other hand, are less common in IoT devices. Bluetooth 5.1 supports native AoA estimation with a single radio front-end [12]. Thanks to its low cost and low power consumption, Bluetooth is widely used in

Table 1. Comparison of BLEselect and other HMD-based IoT device selection methods.

Signal	*HMD Hardware*	*IoT Hardware*	*Low Power (HMD)*	*Low Power (IoT)*	*Privacy-preserving*	*Head Gesture*	*Hand Gesture*
Acoustic [42]	Microphones	Speaker	✗	✗	✗	✓	✗
Infrared [48]	IR receiver	IR transmitter	✗	✗	✓	✓	✗
Visual [4, 20, 22, 24, 32, 46]	Camera	None/barcodes/LED	✗	✗	✗	✓	✓
BLEselect	**BLE receiver, antenna array**	**BLE transmitter**	✔	✔	✔	✔	✔*

* with a wrist-worn BLE transmitter (*e.g.*, a smart watch)

IoT devices and mobile devices like IoT sensors, smart speakers, and smartphones. Bluetooth also supports ad-hoc AoA measurements of nearby devices since it does not rely on existing infrastructures like WiFi. It consumes much less energy (10-20mW) when compared with UWB (500mW while listening [8]) and mmWave FMCW radar(148mW for the an 24GHz low-power radar transceiver MMIC [4]). BLEselect's AoA estimation performance is limited due to the constrained power, size, and hardware resources. However, we show in later sections that such an AoA estimation accuracy is sufficient for IoT device selection tasks.

2.3 IoT Device Selection Methods

As the number of IoT devices increases in our daily lives, researchers have proposed various selection methods using different signals. Users can tap a smartphone [43] or a smart ring [51] on an IoT device, shake two devices simultaneously [26], and tap [49, 50] or move [38] in synchronous with the target device for selection purposes. Such methods require users to contact the target IoT device, though.

For remote selection, users can point a smartphone at [6, 29, 31, 37] or take a picture of the target device [10, 13]. When using a hand-held AR display, the locations of IoT devices can be rendered in real-time, which affords easy selection by clicking on the screen [19, 35]. Different signals can also be fused together to improve selection accuracy [3, 5, 17, 22]. Such methods, however, either require extra hardware (*e.g.*, infrared transceivers) or rely on environmental infrastructure (*e.g.*, WiFi router and RFID reader). The visual signals used also increases the chances for user privacy infringements.

As head-mounted computing devices like AR and VR headset gains traction, researchers start to leverage such HMDs for a user-centered IoT device selection experience. HOBS [48] adds IR transceivers on both the HMD and the IoT device for head orientation-based selection in physical spaces. FaceOri [42] leverages the microphones inside noise-concealing earphones to estimate the direction of incoming acoustic chirps. However, the required hardware (IR transceiver, speaker) adds cost and power consumption to the resource-constrained IoT devices. By capturing images of the surrounding area, HMDs can locate IoT devices by object recognition [20, 24], localizing the attached barcodes [32] or RFID tags [22], and even decoding the blinking of the device's LED [4, 46]. Users can then select the target device through dwell or spatial [34]/ temporal [49] correlations of gestures or gaze. Video recording can violate user privacy, which is especially true for HMDs. Both the IR transceiver and the camera can quickly drain the HMD's battery. BLEselect only uses Bluetooth radio chips on both the HMD and IoT devices, which saves both cost and power on such devices. It works with Bluetooth 5.1 compatible IoT devices without any hardware modification. We show the differences between BLEselect and other HMD-based device selection methods in Table 1.

[4] https://www.infineon.com/dgdl/Infineon-BGT24LTR11N16-DataSheet-v01_03-EN.pdf?fileId=5546d4625696ed7601569d2ae3a9158a

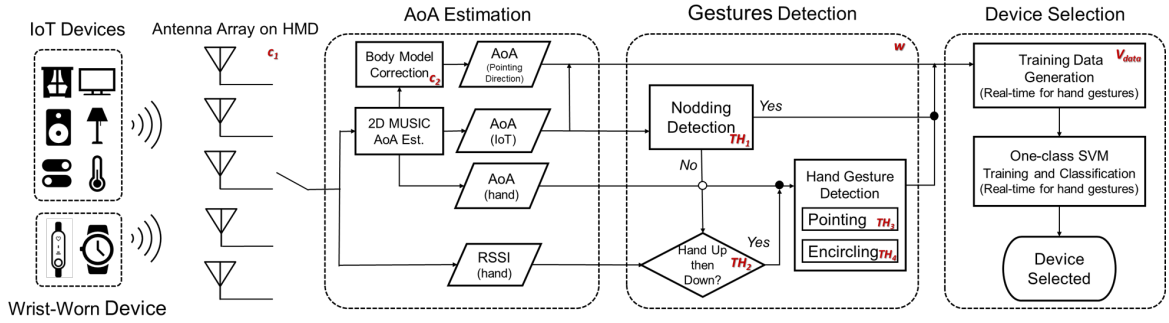

Fig. 3. BLEselect System Overview. System parameters are marked in red.

3 METHOD

The BLEselect system overview is shown in Figure 3. BLEselect adds an antenna array on the smart glasses to estimate the AoA of surrounding Bluetooth 5.1 compatible devices, which is used to detect gestures and select the target device. Details on the antenna array design can be found in Section 3.1. The captured data are fed into the 3-step system pipeline: AoA estimation, gesture detection, and device selection. The system first estimates the AoA of both IoT devices and the wrist-worn device. It uses a 2D MUltiple SIgnal Classification (MUSIC) algorithm to estimate the AoA of incoming signals. Then A human body model transforms the AoA of the wrist-worn device to the finger-pointing direction. AoA estimation details can be found in Section 3.2. Based on the AoA information, the system recognizes the three types of gestures. For the head *Nodding*, it monitors the averaged AoA standard deviation (STD) of all tracked devices; For the hand *Pointing* and *Encircling*, it monitors the RSSI of the wrist-worn device to detect hand up and down events, then analyzes its AoA STD and FFT to recognize the two gestures, respectively. Gesture detection details can be found in Section 3.3. After the gestures are detected, the system augments the AoA data and trains machine learning models in advance for *Nodding* and in real-time for *Pointing* and *Encircling*. Such models determine which IoT device the user intends to select through the gesture. Device selection details can be found in Section 3.4. The various system parameters are marked in red in Figure 3. Their functions and setting rationales are explained in corresponding sections and summarized in AppendixB. The parameters' generalizability across users and devices are discussed in Section 6.

3.1 HMD Antenna Array Design

The receiver on the smart glasses consists of a Bluetooth 5.1 compatible transceiver, an RF switch, and an antenna array. The antenna array is the critical component that greatly impacts on the AoA estimation performance. The overall performance of an antenna array is determined by two factors: the type of the antenna elements and their relative positions. The goal is to design an antenna array that fits onto the frame of a pair of smart glasses and has a high AoA estimation resolution (*i.e.*, smaller resolvable distance) in both the azimuth and the elevation plane.

3.1.1 Rule of Thumb between Resolution and Antenna Array Aperture. The resolvable distance of a receiver on any plane (*e.g.*, the azimuth and the elevation plane) can be roughly estimated by the following equation [28]

$$d \approx \frac{\lambda R}{A} \tag{2}$$

in which d is the resolvable distance in the plane, λ is the wavelength, R is the range between the receive and the target, and A is the aperture size on that plane. Clearly, the larger the antenna array size on that plane, the better the corresponding resolution.

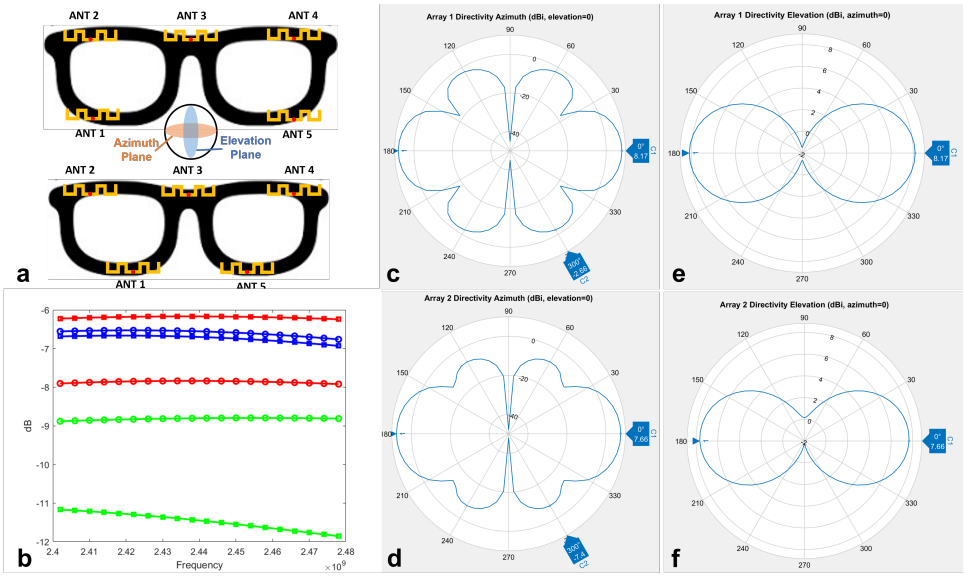

Fig. 4. a) Array 1 (top) and Array 2 (bottom); b) Simulation results of cross-coupling S12(red), S23(blue), S13(green) for both Array 1 (square marker) and Array 2 (circle marker); c) Array 1 directivity on the azimuth plane; d) Array 2 directivity on the azimuth plane; e) Array 1 directivity on the elevation plane; f) Array 2 directivity on the elevation plane.

3.1.2 What is the Distance D between Adjacent Antenna Elements? To avoid AoA estimation ambiguity, the distance between adjacent antennas should be less than half of the shortest wavelength in the communication band, which is 60.5mm at the 2480MHz Bluetooth channel, *i.e.*, $D < 60.5$. At the same time, D should be large to realize a large antenna aperture and minimize cross-coupling between antenna elements. Thus in our design, we set $D = 60mm$.

3.1.3 What is the Overall Size of the Array? Ideally, we would want the size of the array to be as large as possible (*i.e.*, with many antenna elements) for high resolutions in both the azimuth and the elevation plane. However, the array length and height are constrained by the glasses' frame. In this paper, we set the length of our prototype glasses frame based on the human face width, which is 145mm. This is the average value of the 95th percentile of the bizygomatic face breadth of both men and women [2]. The height of the frame is set to D to afford an antenna array aperture as large as possible on the elevation plane while maintaining a typical appearance of glasses. To sum up, the antenna array has a width $W = 145mm$ and a height $H = 60mm$.

3.1.4 Which Type of Antenna to Use as the Array Elements? Antenna sizes, shapes, directivity, and efficiencies are key factors to consider for an antenna element. The antenna should have a length smaller than D, a height comparable to the frame width (8mm), with a small or no grounding area. The antenna should be low-profile to avoid extrusion from the glasses frame. The antenna efficiency should also be high to save power for the HMD and avoid generating heat that makes the user uncomfortable. Its directivity should have a null along the frame plane to minimize cross-coupling in the azimuth direction. One widely used type of antenna that meets all the above requirements is the dipole antenna, which has a length of exactly D and a high efficiency. However, a length of D is still too large since the nose will block the antennas along the lower frame. Thus we use meander dipoles as array elements, which have a shorter length but only a slightly lower efficiency than a dipole antenna.

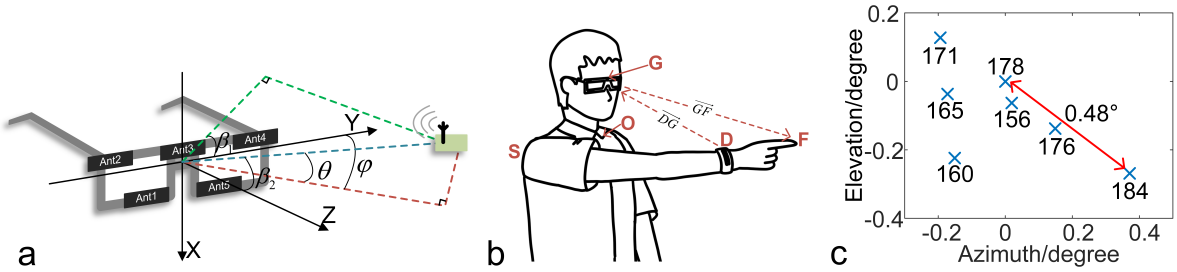

Fig. 5. a) Relative position of the glasses to the device for calculating the AoA. Blue line is the direction of device. Green line is the projection on the XY plane and red line for the projection on the YZ plane; b) Diagram of human body model transforming with wrist-worn device; c) Direction of pointing using standard human body model (178cm) when people of different heights all point at the center direction of glasses.

3.1.5 How to Place Each Antenna Element? Based on the above analysis, we can have three antennas placed along the frame length ($(N_a - 1) \times D < W$) and two placed along the frame height ($(N_e - 1) \times D < H$). The top row of three antennas are placed on the top frame with an equal space of D with one antenna in the middle of the frame. However, the bottom row of two antennas can have different placement strategies. Figure 4 shows two antenna array structures, the left one has a larger array aperture in the azimuth direction but suffers from stronger cross-coupling in the elevation direction compared with those of the right one. We conduct a MATLAB simulation to understand the directivity and cross-coupling between different antenna elements.

3.1.6 Antenna Array Directivity and Cross-coupling Simulation. We use MATLAB to simulate two antenna arrays with different antenna placement strategies on the bottom row (Figure 4a). In *Array 1*, the two antennas of the bottom row are placed as far from each other as possible to provide the largest aperture on the azimuth plane; In *Array 2*, the two antennas of the bottom row are placed in a staggered manner with respect to the antennas of the top row to avoid cross-coupling. We use the default meander-line antenna generated by MATLAB at a design frequency of 2450MHz. Figure 4b shows the simulated cross-coupling between antenna elements 1, 2, and 3 (other S parameters are not shown due to symmetry). The cross-coupling between antennas on the elevation direction (S12) of Array 2 is only 63% of that of Array 1. The nulls of the meander antenna reduce the cross-coupling between antennas in the azimuth direction to only around $-6.6dB$ for both arrays.

In terms of directivity, Array 1 has a higher directivity ($8.17dBi$) on the azimuth plane than that of Array 2 ($7.66dBi$). However, Array 1 also has side lobes ($-2.66dBi$) 3 times greater than those of Array 2 ($-7.4dBi$). The higher side lobes will introduce stronger interference from directions that the user is not facing, deteriorating the AoA estimation performance. Aside from the maximal directivity, the differences of directivities on the elevation plane are mainly at the 90°and 270°direction, which align with the user body and do not have a large impact on AoA estimation. So we use Array 2 in our final prototype based on the simulation results.

To sum up, we use two rows of meander dipole antennas in our array design. The first row consists of three antennas equally spaced with a distance of 60mm on the top frame, and the second row consists of two antennas placed 60mm apart in a staggered manner with respect to the top-row antennas on the bottom frame.

3.2 Device AoA and Pointing Direction Estimation

3.2.1 2D MUSIC AoA Estimation. The straightforward calculation of AoA based on phase differences (Section 2.1) is vulnerable to the multi-path effect, polarization mismatch, *etc.* [47] MUSIC Algorithm is one of the most popular radio direction finding algorithms. It decomposes the signal into two orthogonal spaces: the signal space and the

noise space. Note that it is unlikely that two Bluetooth CTE signals will impinge on the antenna array at the same time since 1) the CTE has a short duration (160us maximum) compared with the advertising interval (20ms minimum); 2) the CTE advertising frequency channel is randomly selected. So we can safely assume there is only one CTE signal impinges on the antennas at any time (**s**).

We can model the received signal $\mathbf{x}(t)$ as followed,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \tag{3}$$

where $\mathbf{s}(t)$ is the source signal vector, and $\mathbf{n}(t)$ is a Gaussian noise term with a variance of σ_N^2, $\mathbf{A}$ is the ideal steering matrix. For each antenna, received signal $x_i(t) = \cos\alpha_i(t) + j\sin\alpha_i(t)$ and steering factor $a_i(t) = e^{j\frac{2\pi}{\lambda}(d_{x_i}\sin\beta_2\cos\beta_1 + d_{y_i}\sin\beta_2\sin\beta_1)}$, in which (d_{x_i}, d_{y_i}) is the position of each antenna on the glass in the XY plane, β_1, β_2 is the projected angle in the XY plane and the direction angle with Z axis respectively (Figure 5).

The covariance matrix R_{xx} of $\mathbf{x}(t)$ can be roughly estimated by the time average of $\mathbf{x}(t)$, which is then decomposed based on eigenvectors. The eigenvector corresponding to the highest eigenvalue is assumed to be the signal space, while other eigenvectors are considered to be the noise space. A pseudo spectrum is then constructed based on the steering vector and the noise vector, whose peak represents the incoming direction of the received signal in β_1, β_2. Then, we calculate the azimuth angle $\phi = \arctan(\frac{\cot\beta_2}{\cos\beta_1})$ and elevation angle $\theta = \arctan(\frac{\sin\beta_1}{\cos^2\beta_1 + \cot^2\beta_2})$. A more comprehensive description of the 2D MUSIC AoA estimation method can be found in Appendix A.

3.2.2 Pointing Direction Estimation. A human body model transforms the AoA of the wrist-worn device to the pointing direction (see Figure 5b) We set the coordinate origin at the user's chest. G, S, D, F and F is the position of the glass, shoulder, wrist-worn device, and finger. We measure the body dimensions of a male with an age of 21 and a height of 178cm and build a correction model to estimate the pointing direction for all users in later experiments. The data includes $||\vec{OS}|| = 18cm||$ (half of the shoulder width), $\vec{SF}|| = 66cm$ (arm length), $||\vec{DF}|| = 20cm$ (length of finger and hand), and $||\vec{OG}|| = 20cm$ (height of eyes). We assume that $\vec{GO} \perp \vec{OS}, \vec{OS} \perp \vec{SF}$ and $\vec{SD} \parallel \vec{SF}$. The estimated direction of wrist-worn device D_h is the orientation vector of $\vec{GD}$. So the system decides the position of D on the line with orientation of D_h such that $||\vec{SD}|| = ||\vec{SF}|| - ||\vec{DF}||$. Then we get the orientation $\vec{OF} = \vec{OS} + \frac{||\vec{SH}||}{||\vec{SD}||}\vec{SD}$, which is the actual direction of where user are pointing at. The pointing direction of the user's finger D_f is then the orientation vector of $\vec{GF}$.

To understand the impact of different human body dimensions on estimated pointing direction, we collected human data from 6 more people with heights from 155cm to 185cm and asked them to point at an IoT device at $\theta = \phi = 0$. Figure 5c shows the estimated pointing directions for each participant corrected by the previous model. The largest error is only 0.48°, which occurs for the 184cm participant. The results show that the correction model does not introduce large estimation errors for people with different heights.

3.3 Selection Gesture Design and Recognition

3.3.1 Selection Gesture Design with a User Study. We designed three selection gestures which users are already familiar with: 1) the dynamic *Nodding* head gesture: the user nods at the target device just as how people acknowledge each other; 2) the static *Pointing* hand gesture: the user points at the target device and dwell for a while; 3) the dynamic *Encircling* hand gesture: the user draws a circle in the air around the target device. Users can nod several times (N_d), hold the pointing for several seconds (T_p), or draw multiple circles (N_c) to improve selection accuracy.

To find the user preferred gesture parameters, We conducted a user study with 18 participants (6 females) from our institution with ages from 22 to 42 (MEAN=26.7, SD=5.56). The experiment lasted for about 15 minutes, and

each participant was compensated with 3 USD for their time. We asked the participant to sit in front of a working desk and perform the gestures to pretend to select one of the five objects: a refrigerator to the left, a headset to the right, a trash can to the bottom, a light to the top, and a monitor in front of the user. All objects are within 2 meters from the user.

There were nine sessions, each corresponds to a gesture configuration (*Pointing* with $T_d = 1, 2, 3s$, *Encircling* with $N_c = 1, 2, 3$, and *Nodding* with $N_d = 1, 2, 3$). Within each session, the participant selected each object in a random order using the same gesture with a 2 seconds rest between consecutive selections. After each session, participants rated their preferences of the gesture configuration on a 7-point Likert scale, with 7 being *Extremely easy*, 4 being *Barely acceptable*, and 1 being *Not acceptable*. The sequence of sessions is counter-balanced using Latin Square.

The results show that users prefer *1-second Pointing* (MEDIAN=6) to *1-time Encircling* (MEDIAN=5.5), though a Wilcoxon test does not show significant rating difference ($W = 28.5, p = .46$). In fact, 9 out of the 18 participants pointed out that they prefer *Encircling* to *Pointing* in the exit interview since *Encircling* can "designate the target device more precisely" and "feels easier than holding the hand still in the air". Two participants mentioned that they would use nodding when it is awkward to use hand gestures (*e.g.*, on public transportation, during a meeting). To sum up, users prefer using *1-second Pointing* and *1-time Encircling* to select an object if possible. *1-time Nodding* is also preferred under specific scenarios. So we use the user-preferred three gestures in later selection evaluations.

3.3.2 Gesture Recognition Pipeline. We develop a gesture recognition pipeline (Figure 3) to support all three gestures so that users can choose their preferred selection gesture that best suits the context. The pipeline uses a 0.8 second time window for gesture detection. It determines whether the user's head is still by analyzing the variances of estimated AoA of the IoT device closest to the origin point (*i.e.*, the device that the user looks at). A large variance of the elevation angle ($\sigma_{elevation} > 28^\circ$) with a small variance of the azimuth angle ($\sigma_{azimuth} < 4^\circ$) indicates the start of the *Nodding* gesture. The thresholds are empirically determined. When the $\sigma_{elevation}$ becomes small again and the value of the azimuth angle is similar to that before the nodding starts, the system marks the end of the nodding gesture.

If the head is still, the pipeline continues to detect whether the hand is raised by monitoring the wireless signal strength of the wrist-worn device through the Received Signal Strength Indicator (RSSI) specified by the Bluetooth protocol. The *hand-up* event leads to a significant RSSI value increase while the *hand-down* event introduces a great RSSI value decrease, which marks the start and the end of the hand selection gesture. We set a 0.4s time window with a moving average filter for this detection. The thresholds of beginning and end are set to be $-44dbm$.

Between the *hand-up* and the *hand-down* events, when the average value of the converted AoA of wrist-worn device satisfy $|\phi| < 25^\circ, |\theta| < 25^\circ$, the system will proceed to the next step of gesture detection. Otherwise, the system will label the behavior as a random hand movement. For hand gestures recognition, a *Pointing* gesture is detected if the standard deviation of the wrist-worn device is less than a set threshold (3°) for a period. The *Encircling* gesture, on the other hand, creates sinusoidal changes in the azimuth and elevation angles. We apply a 64 points Fast Fourier Transform(FFT) to both the elevation and the azimuth angles. An *Encircling* gesture is detected if frequency components less than 0.5Hz in either the azimuth or the elevation plane have magnitudes larger than 4.5.

Figure 6a-c shows the time-domain AoA signals of three gestures, and the orange dash box denotes the gesturing period. The first two rows are estimated azimuth and elevation angles, while the third row is RSSI values for hand gestures. *Encircling* angle signals are filtered with a Kalman Filter (yellow line). The 10-point average of the RSSI values are also shown (red lines).

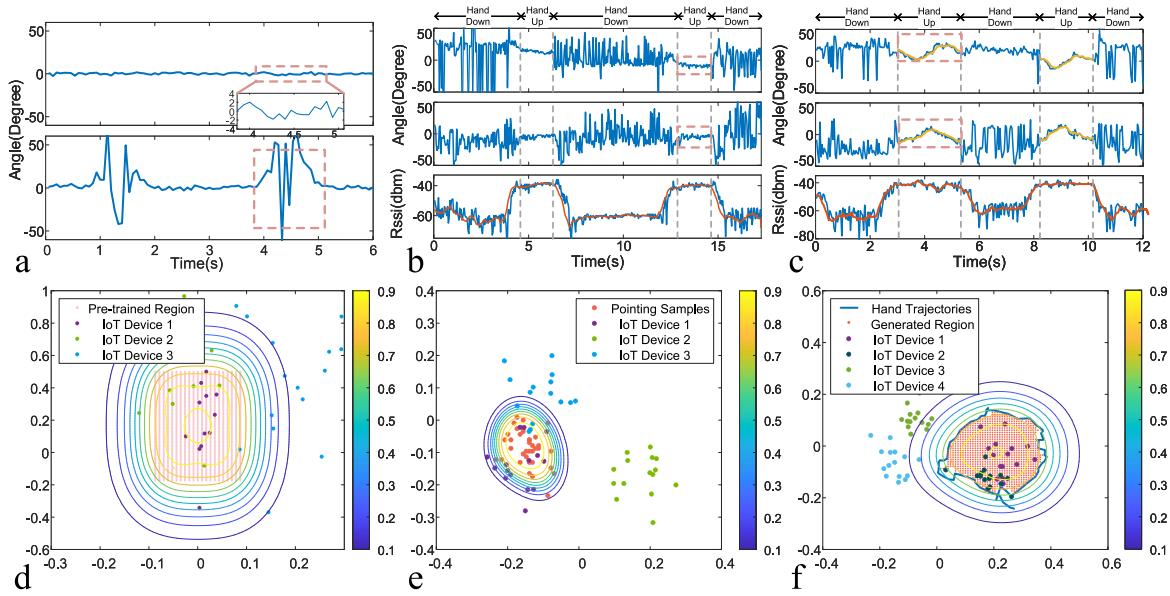

Fig. 6. a) AoA of IoT Device with *Nodding* Gesture; b) AoA of wrist-worn Device with *Pointing* Gesture; c) AoA of wrist-worn Device with *Encircling* Gesture; d) SVM Prediction of *Nodding*; e) SVM Prediction of *Pointing*; f) SVM Prediction of *Encircling*.

3.4 Target Device Selection Algorithm

The system then matches the gesture with one of the IoT devices for device selection. We tested two methods for target device selection in this paper. The Angular Distance Method relies on the AoAs of IoT devices and their relative angular differences from the pointing direction to complete selection. The Machine Learning Classifier-based Method, on the other hand, trains ML models that take AoA of an IoT device as input and output whether it is the selection target.

3.4.1 Angular Distance Method. We quantify the 3D angular distances on the azimuth and the elevation plane separately since our antenna array estimates the azimuth angle more accurately. The distance metric places more weight on the azimuth angle than the elevation angle, which is shown below.

$$\Delta E(D_h, D_{dev}) = -(a_1 \sin(\theta_h)\sin(\theta_{dev}) + a_2 \cos(\theta_h)\cos(\theta_{dev})\cos(\phi_h - \phi_{dev})) \tag{4}$$

where D_h, D_{dev} represent the pointing direction and the AoA of surrounding IoT devices respectively with their elevation and azimuth component $\theta_h, \theta_{dev}, \phi_h, \phi_{dev}$. The weights a_1, a_2 are empirically set to be 1 and 1.5.

Then system selects the target device i^* by solving the following equation

$$i^* = \arg\min_i \left(\sum_{j=1}^{N} \Delta E(D_h(j), \overline{D}_{dev})\right) \tag{5}$$

where $D_h(j)$ is the j^{th} sample of wrist-worn device direction during the selection period. $\overline{D}_{dev_i}$ is the estimated AoA of the i^{th} IoT device averaged between the start and the end of the gesture. For *Nodding* selection, both the azimuth and the elevation angels of the pointing direction D_h is set to 0°.

3.4.2 ML Classifier-based Method. We can also treat the IoT device selection task as a classification problem. Each IoT device is classified by a model generated with gesture data to determine if it is the selection target.

We first apply trigonometric mapping to improve the model's sensitivity when the azimuth and the elevation angle are near zero. We then train machine learning classifiers for each performed gesture instance separately and use such models to determine whether an IoT device is the selection target. The algorithm1 shows the selection procedure.

Trigonometric Mapping When using BLEselect, users will look directly at the target IoT device during the selection period, which orients the antenna array so that the azimuth and elevation angles of the target device are both close to zero. So we apply a trigonometric mapping $(a, e) = (sin\phi, sin\theta)$ for all AoA data to improve sensitivity around such a region.

Training Data Generation We first filter out the outline data points that have a larger than three scaled median absolute deviations (MAD) from the median in either the azimuth or the elevation plane (Scaled MAD is defined as $c * median(abs(A - median(A)), c = -1/(\sqrt{2} * erfcinv(1.5)))$. 1) For *Nodding*, we pre-train a classifier using generated AoA data within the region of -5°< Azimuth < 5°and -10°< Elevation < 30°with a 0.5°step size (same step to search the peak of spectrum in the 2D-MUSIC Algorithm). This region is decided through a pilot study. We ask four people to nod at an IoT device and found that a majority of the AoA data of the IoT device falls into this region. 2) For *Pointing*, we train a machine learning model in real-time for each selection gesture, right after the detection of a *hand-down* event. We extract the AoA data of wrist-worn device within the period from *hand-up* to *hand-down*, which approximates the AoA of pointing target. So we use such data (orange dots in Figure 6e) to train a new model for each *Pointing* gesture. 3) For *Encircling*, similarly, we train a new model for each gesture in real-time. We first apply a Kalman filter($Q = 10^{-4}, R = 4 \times 10^{-5}$) to smooth the moving trajectory of the pointing direction. Then we fill the region enclosed by the drawn circle (blue line in Figure6f) with a 0.5°step size both in the azimuth and the elevation plane.

Model Prediction and Device Selection For each IoT device, we feed its recorded AoA data points during the selection period (within the orange dash box in Figure6a-c) into the model and calculate its average credibility. The IoT device with the highest average credibility is then selected. Three examples are shown in Figure 6d-f. The color contour shows the credibility in one specific region. The XY axis are mapped from (ϕ, θ) to $(\sin\phi, \sin\theta)$. IoT device 1 is the target device in all three subfigures.

Algorithm 1 Machine Learning based Selection Algorithm

Input: AoA of IoT Devices (ϕ_j, θ_j),
Selection Gestures $SeleGes \in \{Pointing, Encircling, Nodding\}$,
AoA of wrist-worn Device (ϕ_h, θ_h) (if hand-selection ensured)

$(a_j, e_j) \leftarrow (\sin\phi_j, \sin\theta_j)$, $(a_h, e_h) \leftarrow (\sin\phi_h, \sin\theta_h)$
if $SeleGes \in \{Pointing, Encircling\}$ **then**
  $(a_h^{'}, e_h^{'}) \leftarrow$ AoAmodify$((a_h, e_h), SeleGes)$
  $Model \leftarrow$ Modeltrain$(a_h^{'}, e_h^{'})$
else
  $Model \leftarrow$ Modeltrain$(PreTrainedRegion)$
end if
$llr_j \leftarrow$ Modelprediction $(Model, (a_j, e_j))$
$i^* \leftarrow$ argmax (mean(llr_j))

Output: Target Device i^*

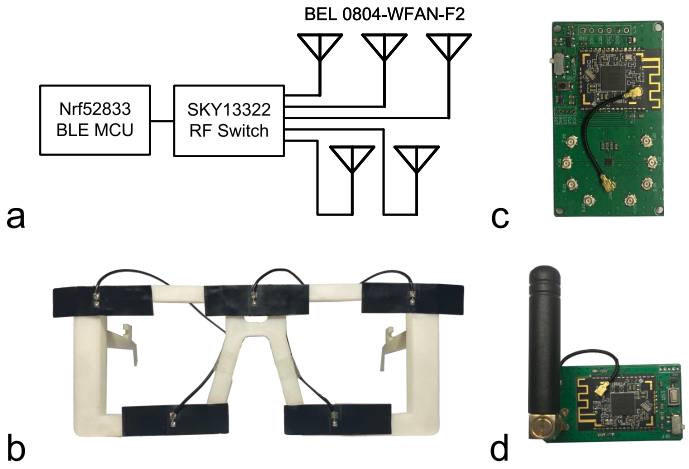

Fig. 7. a) Receiver block diagram; b) *BLEselect* smart glasses with the antenna array; c) Receiver PCB board with IPEX connectors for antennas; d) Transmitter as the IoT device.

In Section 5.2.2, we compared performances of 4 machine learning classifiers (Naive Bayes, KNN, Random forest, SVM) through an experiment. The results show that the one-class SVM has the best overall performance in terms of selection accuracy and training time, which is used in later selection experiments and use studies.

4 IMPLEMENTATION

We 3D printed a frame of glasses with a width of 145mm and a height of 60mm, on which we deploy our receiver.

4.1 Hardware

4.1.1 Receiver. The receiver consists of a Bluetooth 5.1 compatible MCU, an RF switch, and an antenna array. In our prototype, We use a PTR9813 Bluetooth module[5] with Nordic nRF52833[6] wireless MCU to scan for Bluetooth advertisements. We use SKY13418[7] as the RF switch that connects the RF port of the Bluetooth chip to the five antennas. The receiver circuit board has a dimension of $3.4cm \times 5.6cm$ and is powered up by a CR2032 battery. To ensure robust RF performance and better reproducibility, we use the COTS BEL 0804-WFAN-F2 FPC antenna as the array element, which is inferred to be a meander-line dipole antenna based on its size ($40mm \times 12mm$) and 2D radiation pattern [8]. It has a high efficiency of 75% within the Bluetooth band at 2440MHz. We connect the antennas with the receiver board using IPEX connectors. The antennas are attached to the glasses' frames without occluding the sight. The receiver can report the phase data to a PC either through wired UART port or a wireless 433MHz data transfer module.

4.1.2 Transmitter. We use the same PTR9813 module and a 3cm whip antenna for the transmitter. The advertising frequency is 12.5Hz for IoT devices and 25Hz for the wrist-worn device to capture fast hand movements. A CR2032 coin battery powers the transmitter.

[5] http://en.abluetech.com/plus/view-1135-1.html
[6] https://www.nordicsemi.com/products/nrf52833
[7] https://www.skyworksinc.com/Products/Switches/SKY13418-485LF
[8] https://1drv.ms/b/s!AqMWOjbknXwDiYlb4PVOa897xUGjUw?e=lrKrcC

4.2 Software

We calculate and interpolate the phase data on-chip based on the measured I/Q data of the CTE packets from each antenna element. The phase data and RSSI data from each CTE packet are transmitted to a laptop (Intel i7 9750h CPU, 16GB RAM) via a serial port or a 433MHz wireless communication module. The MATLAB program reads all serial port data for later processes. We implement the AoA estimation algorithm to both the IoT devices and wrist-worn device (with hand model correction). The calibration module applies linear calibration to the phase data of antennas based on the previous measurement results. Next, the program detects *Nodding* using AoA data of IoT devices. It completes a selection according to the ML model prediction with a pre-trained SVM model if *Nodding* is detected. If not, the program starts to look for *hand-up* to *hand-down* events. It then determines what kind of gestures is performed (*Pointing*, *Encircling*, irrelevant gestures). After a hand selection gestures are detected, the augmented data of the wrist-worn device are loaded into a ML training module to train a real-time one-class ML classification model. The AoA data of broadcasting IoT devices are then fed into the model. Finally, the IoT device with the highest post-probability predicted by the model is selected.

5 EVALUATION

In this section, we first measure the direction-finding errors of our receiver. Then we perform micro-benchmark tests to characterize four critical parameters of our system (supported device number, ML model, spatial resolution, power). Next, we examine the selection accuracy of the three gestures in three different environments. At last, a user study is conducted in two settings to evaluate the performance of our entire system both objectively and subjectively.

5.1 AoA Estimation Accuracy

The selection performance of our system is highly dependent on the AoA estimation accuracy. We conduct an experiment in a teaching building hall with an open area of $8.5m \times 5.3m$. The experimenter wore the glasses and the receiver on his head, and stood facing a wall with a distance of 1.5m, 2m, 3m, and 5m, respectively. The location right in front of the center of the glasses is marked with 0°azimuth and elevation angle, which the experimenter faced during the whole experiment. We placed transmitters at 35 locations on the wall with 7 azimuth angles ranging from -30°to 30°and 5 elevation angles ranging from -20° to 20°, both with a spacing of 10°. The advertising frequency for the transmitters is set to 12.5Hz.

Given the actual direction with (θ, ϕ) and the estimated direction with $(\hat{\theta}, \hat{\phi})$, the 2-D direction finding angle error is defined by the angular distance between these two direction value [14]. The absolute angular error in the direction (θ, ϕ) is

$$\Delta e(\theta, \phi) = \arccos(\sin(\theta)\sin(\hat{\theta}) + \cos(\theta)\cos(\hat{\theta})\cos(\phi - \hat{\phi})) \tag{6}$$

Figure 8a-d shows the estimated AoAs and the ground truths of transmitters at different azimuth and elevation angles. The AoA estimation performance of our receiver deteriorates slightly within a 3 meters distance and reaches limits at a 5 meters distance. Figure 8e-h shows that the estimation errors are 2.87°, 4.06°, 6.01°, and 9.64°for 1.5, 2, 3, 5 meters distances respectively. The error increases about 2°-3° from the center ($\theta = \phi = 0$) to the corners $|\theta| = 20, |\phi| = 30$.

Figure 9a-d shows that the AoA estimation errors are less than 10° for distances within 2 meters and reaches 20° at a distance of 5 meters for 80% of measurements. When we limit the azimuth and elevation within -10°to 10°, the error reduces by around 20% for all distances. This indicates that our receiver has better performance if the user faces the target device during the selection. We also post-process the data to understand the AoA estimation performance with three antennas (Ant_1, Ant_3, Ant_5 in Figure 5a). Results show that the 3-antenna configuration significantly increases the average AoA estimation error by 50.6%, 55.8%, 59.6% and 66.4% at 1.5m, 2m, 3m, and 5m, respectively.

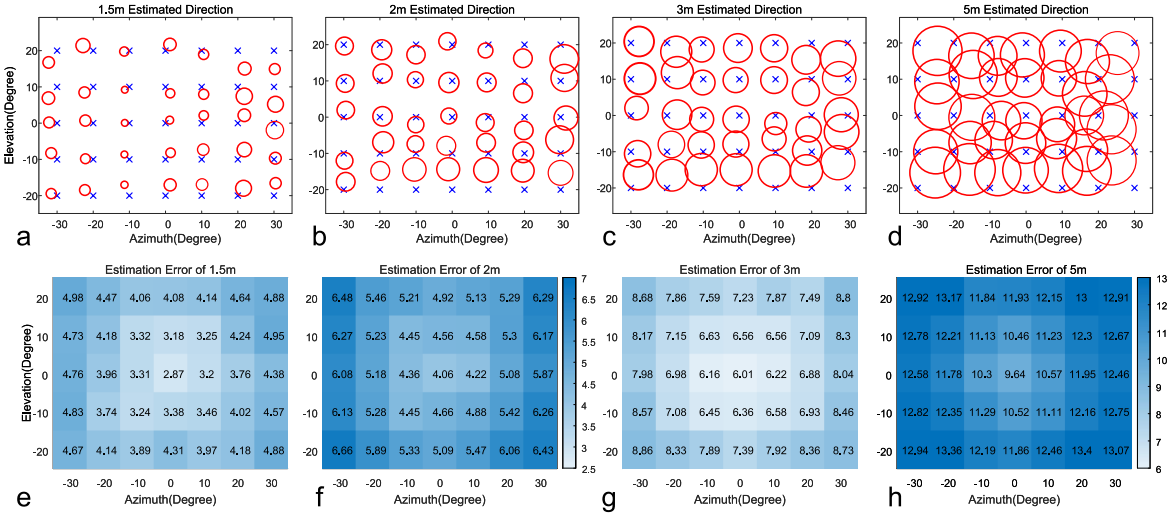

Fig. 8. The actual (blue cross) and the estimated (red circle) AoA at 1.5m(a), 2m(b), 3m(c) and 5m(d). The circle's center is the mean, and its diameter is the standard deviation of estimated 2D angles. Heatmaps of average absolute AoA errors at the four distances are shown in e-h.

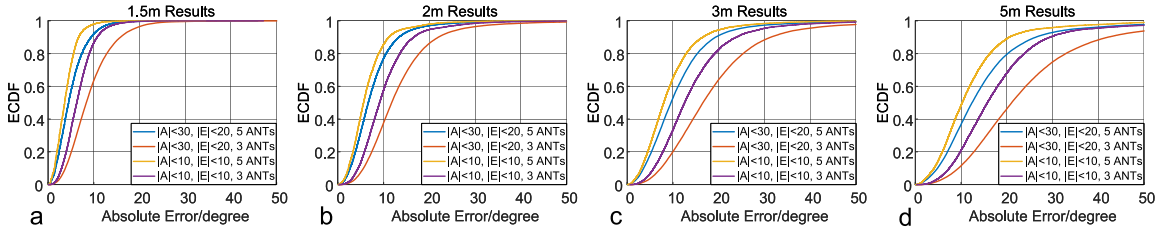

Fig. 9. The ECDFs of the 2D AoA estimation errors at 1.5m, 2m, 3m, and 5m distance for $|Azimuth| \leq 30, |Elevation| \leq 20$ and $|Azimuth| \leq 10, |Elevation| \leq 10$ with 3 and 5 antennas.

It is expected the relatively small-size antenna array worn on the head will have a larger angle estimation error, especially on the elevation plane, as discussed in Section 3.1. So we project the estimated 2D angle to the azimuth and the elevation plane to get the azimuth and the elevation angle estimation error separately. When averaged across all angles, distances, and environments, the azimuth error (MEAN=5.23°) is less than the elevation error (MEAN=6.64°). The result validates our strategy to put a higher weight on the azimuth angle to improve selection accuracy for the Angular-distance based selection method.

We also compared the receiver size, RX power, TX power, and AoA estimation error of BLEselect with those of existing systems in Table 2. We can see that ultrasound, UWB, and IR solutions have rather high power consumptions either on the HMD or on the IoT device. The Zigbee system uses a large COTS receiver with superior RF performance and requires movement of the receiver for AoA estimation. The TI BOOSTXL-AOA Bluetooth 5.1 evaluation kit achieves a lower AoA error since it has a larger receiving aperture with two antenna arrays, each with 3 folded dipole antennas. BLEselect's AoA estimation error is larger due to the size and power

Table 2. Comparison of AoA estimation performance of BLEselect with some existing techniques. We calculate power from the hardware datasheet if it is not provided in the reference.

Signal	*Receiver Size*	*Receiver(RX) Power*	*Tag(TX) Power*	*AoA Error*
Ultrasound [27]	6.5cm x 6.5cm PCB	14mW active	300mW active	~2°at 1m
UWB [9]	2.3x1.3x2.9cm DWM1002	528mW active	462mW active	<5°at 4m
Zigbee+IMU [9]	9.7x15.5cm USRP B210	51mW active	60mW active	~5°at 6m
IR+RF [5]	5x4x3cm	90mW avg	81uW avg	<2°at 9m
Bluetooth 5.1 [1]	19.5x11.3cm BOOSTXL-AOA	20mW active	15mW active	<4°at 2.8m
BLEselect	3.4x5.6cm PCB and 14.5x6cm Ant. Array	<10mW avg	1.9mW avg	3-5°at 1.5m, 6-8°at 3m

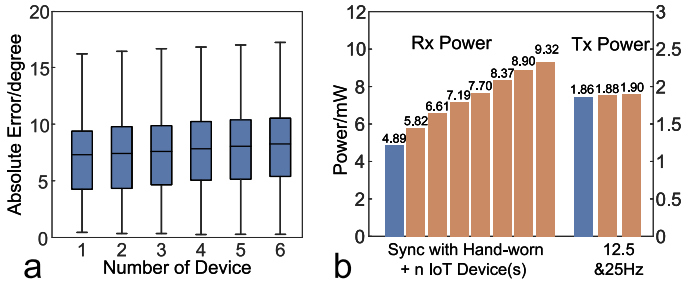

Fig. 10. a) Box-plot of absolute angle estimation error when estimating AoA of n devices simultaneously; b) Power consumption of Rx and Tx. Blue and red represent standby state and working state (Rx:sync with wrist-worn and $n(0-6)$ IoT devices, Tx: advertising frequency at 12.5Hz and 25Hz).

constraints. However, we show in later sections that it is still sufficiently accurate for IoT device selection tasks in our everyday lives.

5.2 Micro-benchmark Experiments

5.2.1 Do More IoT Devices Impact the AoA Estimation Accuracy? Different transmitters may interfere with each other due to CTE collisions. Such interference will reduce the overall AoA estimation accuracy. To quantify the impact, we synced the receiver with 1 to 6 transmitters in an outdoor environment. The distance between the transmitters and the receiver is 2 meters, and the transmitters are placed close together at $\theta = \phi = 0$. One-way ANOVA results show that the number of synced transmitters has a significant impact on the AoA estimation error ($F_{5,27020} = 35.9, p < .001$). However, the average AoA error difference between 1 synced (7.13°) and 6 synced (8.21°) transmitters is only 1.08°(Figure 10a), which translates to less than 4cm at a 2 meters distance. Thus we conclude that the slight AoA estimation error increase due to multiple syncing transmitters is negligible for selection tasks.

5.2.2 Which Selection Algorithm Performs Better? We conducted a selection experiment outdoor to evaluate five selection algorithms: 1) Angular distance; 2) Naive Bayes; 3) K-nearest neighbor(k=10 with euclidean distance); 4)

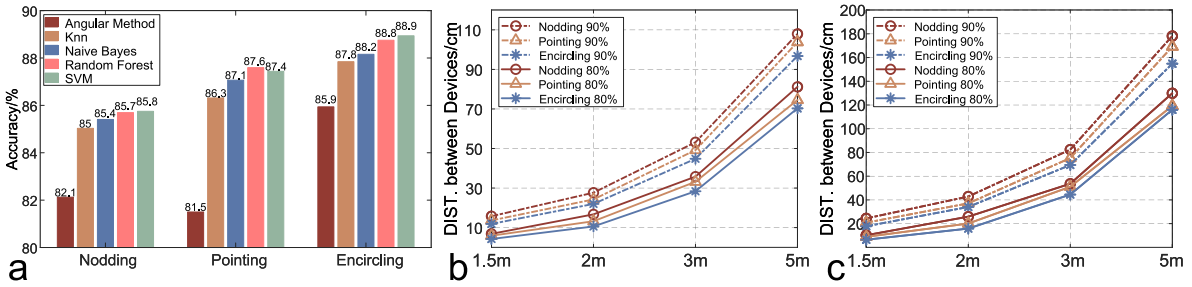

Fig. 11. Accuracy comparison of Angular distance-based method, Naive Bayes, Knn, Random Forest and SVM on selection accuracy of three gestures (a). Resolution of each gesture with different distance and accuracy threshold utilizing SVM-based selection method with 5 antennas (b) and 3 antennas (c). Dash lines represent 90% threshold and solid lines are 80% threshold.

Random Forest($n_{estimators} = 30$); 5) One-class SVM ($C = 0.1, 0.1, 1$ and $\gamma = 0.1, 1, 0.1$ for *Nodding*, *Pointing* and *Encircling* respectively). The experimenters select from two transmitters using the three gestures at 1.5m, 2m, 3m, and 5m distances. We deployed two transmitters along a square diagonal on a wall whose center is $\theta = \phi = 0$. For each distance, the length of the square diagonal changes by angle distances from 2°to 16°with a 2°step. For each device separation, we first place two transmitters on the ends of one square diagonal and select each one 25 times using one gesture. Then we place them on the ends of the other diagonal of the square and also select each one 25 times to eliminate biases. We collect 100 selection trials for each gesture with one separation at one distance and 25 for each corner of the square.

Figure 11a shows that the angular distance based selection method has the worst performance. Among the remaining four ML based algorithms, both the SVM and the Random Forest models perform better than the Naive Bayes and the KNN models. However, since the model is trained for each gesture performance in real-time, the selection algorithm should also have high computational efficiency so that the training and inferring time is minimal. The average training time of the Random Forest Model is 0.11s (on a PC with i7-9750H CPU at 2.6GHz, 16GB RAM), which is more than 6 times larger than that of the SVM model (0.016s). So we choose the one-class SVM-based selection algorithm for our system, which is used in later selection experiments and user studies.

5.2.3 What is the Selection Spatial Resolution? We can also analyze the spatial resolution of our system using the data collected above. We calculate the resolution for the 5-antenna (Figure 11b) and the 3-antenna (Figure 11c) configuration respectively. In both cases, the *Encircling* gesture has the best spatial resolution while the *Nodding* gesture has the worst. With a selection accuracy of 80% as the threshold and 5 antennas used, BLEselect has a spatial resolution of less than 10cm at 1.5 meters and 30cm at 3 meters; with a 90% threshold, the spatial resolution is around 15cm at 1.5 and 50cm at 3 meters. At a distance of 5 meters, the resolution deteriorates to 75cm (80% threshold) and 110cm (90% threshold). With 3 antennas, the resolution deteriorates to 10cm, 20cm, 50cm, and 120cm for 80% threshold, while it declines to 20cm, 40cm, 75cm and 170cm for 90% threshold. The average resolvable distance using 3 antennas increased by approximately 1.55 times compared to that when using 5 antennas.

5.2.4 What is the Average Power Consumption of the Receiver and the Transmitter? We use a DC power supply to apply 3V VDD to the receiver and the transmitter, then measure the current to calculate the power consumption. Figure 10b shows that the receiver only consumes 5.82mW on average when synced with the wrist-worn device and 9.32mW on average when synced with a wrist-worn and 6 IoT devices (7 transmitters in total). The transmitter only consumes 1.9mW on average when advertising at a 25Hz frequency.

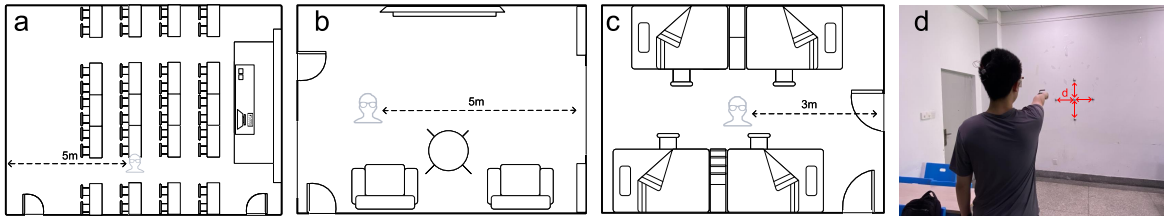

Fig. 12. Floor Plan of Three Environments (a-c) of Selection Accuracy and Experiment Setup (d)

Table 3. Distances between Adjacent Devices in Different Experiment Settings.

	6°	9°	12°	15°
1.5m	15.8cm	23.8cm	31.9cm	40.2cm
2m	21.0cm	31.7cm	42.5cm	53.6cm
3m	31.5cm	47.5cm	63.8cm	80.4cm
5m	52.6cm	79.2cm	106.3cm	134.0cm

5.3 Experiment on Selection Accuracy

We tested the gesture detection and device selection parts of our system in three rooms with different sizes: a classroom (10.8m x 8.2m, Figure 12a), a living room (6.6m x 4.7m, Figure 12b), and a dorm (5.6m x 3.3m, Figure 12c). We expect our system's performance will deteriorate when the room gets smaller due to more severe multi-path effects. We placed five Bluetooth transmitters in a cross shape on a wall of each room, as shown in Figure 12d. The experimenter wore our prototype smart glasses and selected each transmitter using the three gestures. Even though we have shown that our system has the best performance when the user faces the target device, it is difficult to enforce a highly precise orientation of the head. So we ask the experimenter to face the center device for all trials of *Pointing* and *Encircling* to understand the system's ability to tolerate head orientation errors. Target devices are placed 6°, 9°, 12°, and 15°apart from the center device at 1.5m, 2m, 3m, and 5m, respectively (the dorm is too tiny for the 5m testing range). The actual distance d between adjacent devices is shown in Table 3.

Figure 13 shows the accuracy of the selection in different experimental settings. In all three rooms, the *Encircling* gesture has the best performance, which is expected since it provides the richest AoA information. Assuming the user is satisfied with an 85% accuracy, *Encircling* can effectively differentiate between devices that are 15.8cm apart at a 1.5m range, 31.7cm apart at a 2m range, 63.8cm apart at a 3m range, and 134.0cm apart at a 5m range in settings. On average, *Pointing* selection accuracy is 1.0%, 1.125%, and 0.82% lower than those of *Encircling* in the Classroom, Living Room, and Dorm respectively. As the range gets greater, however, the performances of the two hand gestures converge.

To compare selection performances with 3-antenna and 5-antenna array configurations, we average the selection accuracies at different separation angles and ranges across all environments (Figure 14). The selection accuracy of 5 antennas outperforms 3 antennas by 4.6%, 6.75%, 10.68% and 14.2% at the four distances, respectively. The result shows that a larger antenna array aperture can significantly improve selection performances at longer ranges among closer IoT devices.

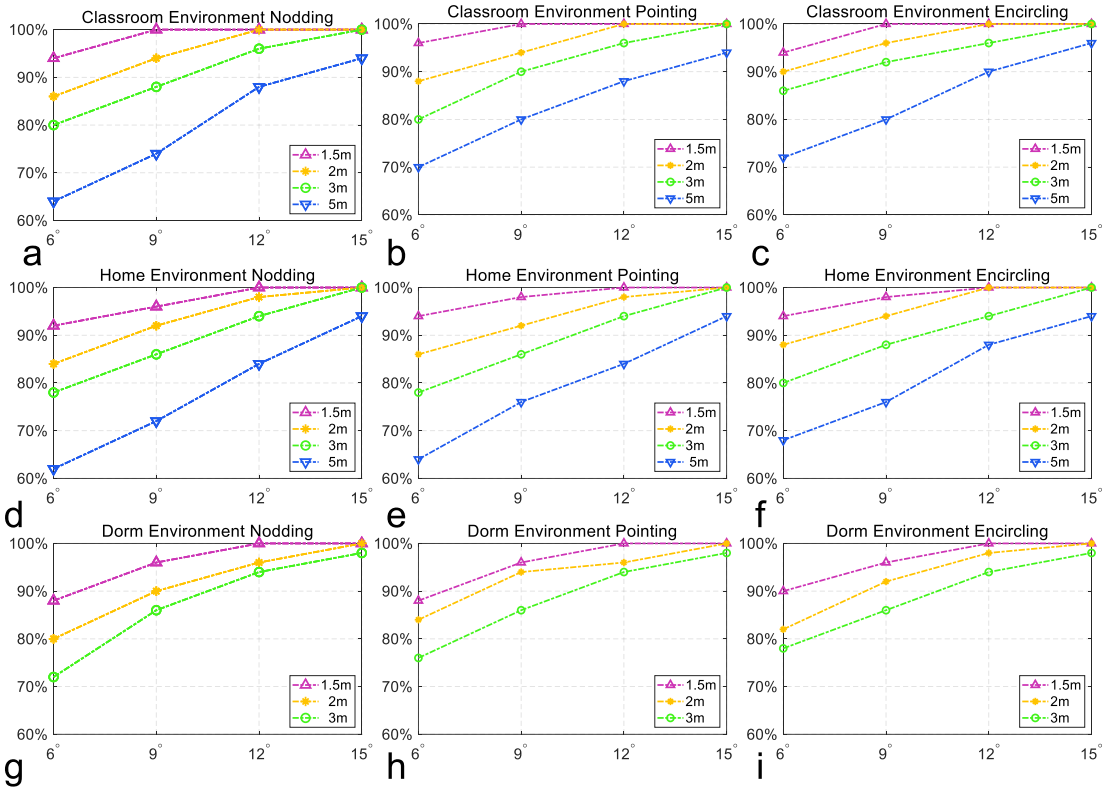

Fig. 13. Selection accuracy at Classroom (a-c), Living room (d-f), and Dorm (g-i) at 1.5m , 2m, 3m, and 5m.

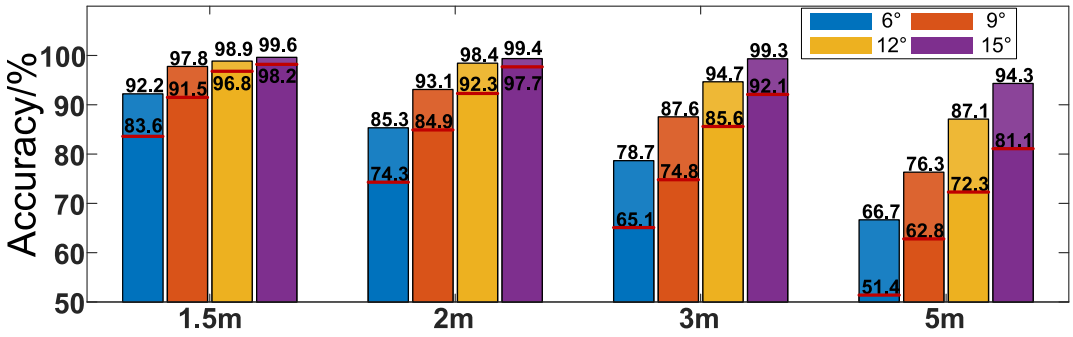

Fig. 14. Comparison of Selection Accuracy between 5 antennas and 3 antennas (red line) for different ranges and separation angles.

Fig. 15. IoT Devices setup for a) dorm and b) classroom environment

.

5.4 User Study

The goal of the user study is to validate that our BLEselect system supports accurate and spontaneous IoT device selection in real-life scenarios. We evaluate our system in the same dorm and the same classroom as in Section 5.3, where users select the transmitter-attached objects using all three gestures. Based on the results of Section 5.3, the two environments allow us to explore the upper and the lower bound of our system's performance.

5.4.1 Participants and Experiment Setting. We recruited two groups of participants separately [9]: 1) 12 university students (9 males) with ages 20-22 (MEAN=20.9, STD=0.67); 2) 10 university staffs (4 males) with ages 47-57 (MEAN=52, STD=3.85). The 22 participants have heights range from 158cm to 185cm (MEAN=170.1cm, STD=8.9cm). They rated their technology savviness themselves on a 5-point Likert scale, with 1 being extremely unfamiliar and 5 being extremely familiar with electronic devices. The self-report ratings of the staff group (MEDIAN=2.5) are much lower than that of the student group (MEDIAN=4). We believe such a diverse set of participants in terms of ages, body structures, and technology savviness can better evaluate our system's generalizability across users.

We attached our Bluetooth transmitters to six objects in each room (distance to the participant in parentheses): *Cabinet1(1.7m), Cabinet2(1.8m), Window(2.4m), Curtain(2.8m), Switch(2.5m), AC(2.9m)* in the dorm (Figure 15a), *Switch1(2.7m), Switch2(3.1m), Blackboard1(2.5m), Blackboard2(3m), Projector(3.1m), Screen(4.8m)* in the classroom (Figure 15b). This arrangement enables us to test BLEselect's performance when transmitters are on objects with different materials and shapes, as well as angles and distances from the user. The participants stand at the fixed location, look at the target device, then perform selection gestures. The data is processed in real-time to output the predicted selection device. Both groups of participants completed the experiment with the same settings and procedures. The experiment lasted 45 minutes, and each participant was compensated with 10 USD.

5.4.2 Experiment Procedure. We adopt a within-subject design for the study. For each room, the participant performed 9 sessions of selection tasks, 3 sessions for each of the *Nodding*, *Pointing*, and *Encircling* gestures. The order of the sessions is balanced with Latin Square to eliminate order effects. Within each session, the participant selected each of the six target objects following a randomized sequence. They rested for 1 minute between sessions. For each selection task, the participant started to perform the gesture when instructed by the experimenter. We recorded three timestamps: the start of the gesture $t_{gesture\_start}$, the end of the gesture $t_{gesture\_done}$, and the end of the processing $t_{calculation\_done}$. $t_{gesture\_start}$ and $t_{gesture\_end}$ are automatically labeled

[9]This is due to a temporary COVID-19 related lockdown of the university.

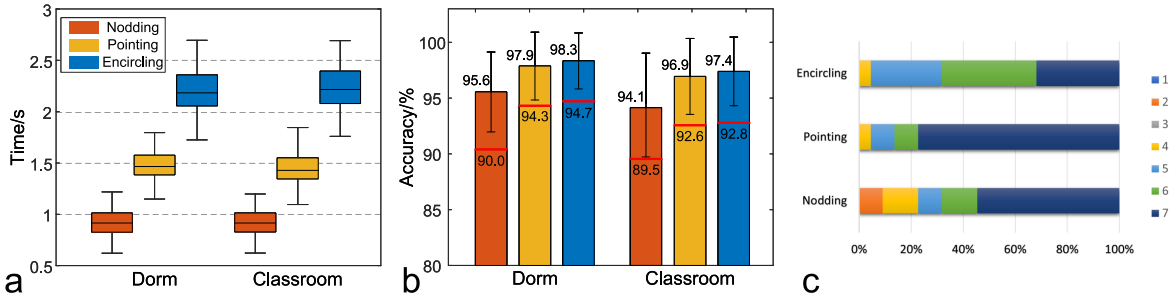

Fig. 16. a) Boxplot of selection time for each gesture; b) Selection accuracy of each gesture (red line indicates the post-processed accuracy with 3 antennas, legend shared with *a*); c) Users' subjective rating of each type of gestures.

using the methods described in Section 3.3.2. At the end of the experiment, the participant rated their overall preferences for each gesture using a 7-point Likert Scale, with 7 being *Highly Preferred*, 4 being *Barely Acceptable*, and 1 being *Totally Unacceptable*.

5.4.3 Results Analysis. We collect data of 22 $participants \times 2$ $environments \times 3$ $gestures \times 3$ $sessions \times 6$ $tasks = 2,376$ selection trials. We calculate the selection time as $t_{calculation\_done} - t_{gesture\_start}$. Figure 16a-b show the selection time and the accuracy, respectively. RM-ANOVA test results with Greenhouse-Geisser corrections show that there is a significant effect of gestures on the selection accuracy ($F_{2,42} = 6.79, p < .01$). Among the three gestures, *Encircling* has the highest selection accuracy (97.8%) but the longest selection time (MEAN=2.2s, STD=0.08s), while *Nodding* has the lowest selection accuracy (94.8%) but the shortest selection time (MEAN=0.91s, STD=0.04s). *Pointing* has an overall selection accuracy of 97.4% with a medium selection time (MEAN=1.45s, STD=0.03s).The difference is minimal for both the overall selection accuracy (Students 96.9%, Staffs 96.5%) and time (Students 1.53s, Staffs 1.51s) between the two groups, which shows our system generalize well across users. The distances between the participant and the transmitters are larger in the classroom than in the dorm, which explains the lower selection accuracy in the classroom. We also monitored whether the gestures would be falsely detected during the whole experiment. During the ~6 hours dorm experiment and the ~7 hours hours classroom experiment periods, there are no false positives for the *Pointing* and the *Encircling* gestures Only 10 and 11 *Nodding* gestures are erroneously detected in the dorm and the classroom, respectively. All of them are detected when the user looked down and up several times to look for the target IoT device. We also post-processed the data to simulate the accuracy with the 3-antenna configuration. There is a 4-5% accuracy drop for each scenario (Figure 16b).

As shown in Figure 16c, the participants highly preferred *Pointing* (MEDIAN=7, STD=0.83) and *Nodding* (MEDIAN=7, STD=1.61), while they also think very positive of *Encircling* (MEDIAN=6, STD=0.88). The results show that our designed gestures are natural to the participants. Wilcoxon signed-rank tests show that *Pointing* has a significantly higher rating than that of *Encircling* ($Z = 10.0, p < .05$). It is more difficult to nod when looking up, which can explain the larger variance of *Nodding* ratings. Two participants echoed that *Nodding* when looking up took too much effort. Several participants mentioned that *Encircling* is more demanding both physically (need to draw a circle in the air) and mentally (not sure whether the circle is drawn correctly), which explains its lower rating. Indeed, in our study *Encircling* and *Pointing* has similar selection accuracies (98% v.s. 97.6%). We believe users will appreciate *Encircling* for its higher selection accuracy when IoT devices are closer together, thus more difficult to differentiate. The user ratings and comments show that users prefer different selection gestures under different scenarios, which validates our system's design to support all three gestures simultaneously.

6 DISCUSSION AND LIMITATIONS

6.1 System Generalizability

We design BLEselect so that it can be used across users, locations, and hardware. We discuss the high generalizability of our system below.

6.1.1 Generalization across Users. To validate our system's generalizability across users, we conducted a user study with a diverse set of participants with vastly different ages, heights, and technology savviness. The 22 participants have ages range from 20 to 57 (MEAN=35.0, STD=16.0), heights range from 158cm to 185cm (MEAN=170.1cm, STD=8.9cm), and self-rated technology savviness range from 1 to 5 (MEDIAN=3.5, STD=1.04). The selection accuracy averaged across 22 users, 2 environments, and 3 gestures is 96.7% with a standard deviation as small as 3.8%. The high overall selection accuracy and the small standard deviation validate that BLEselect has high generalizability across users of different ages, heights, and technology savviness. To further investigate the impact of body structures on pointing direction estimation, we measured body structure data of 6 users and calculate the estimation errors due to using a general human model. The maximal 2D angle error is only 0.48°, which happens for a 184cm user. Such a small error shows that the estimated pointing direction is insensitive to body structures.

6.1.2 Generalization across Locations. BLEselect estimates the AoA of IoT devices relative to the antenna array on the HMD. The estimation accuracy deteriorates at longer distances, larger angles, and more crowded environments. First, there is only a 2°to 3°AoA error increase from the center($\theta = \phi = 0$) to the corners($|\theta| = 30, |\phi| = 20$). Despite the small error, we still ask users to look at the IoT device during the selection so that the device is close to the center of the azimuth-elevation plane. This is natural and ensures the best AoA estimation performance. Second, the distance has a large impact on the AoA estimation accuracy. So we measured the selection resolution at four distances for a selection accuracy threshold of 85%. IoT devices need to be placed 15.8cm apart at 1.5m, 31.7cm apart at 2m, 63.8cm apart at 3m, and 134cm apart at 5m. We want to point out that 5 meters is a very long indoor distance and device selection at such a long distance is relatively rare. The AoA estimation accuracy is also impacted by the signal-to-noise ratio of wireless signals, which can be improved by increasing the link margin (*e.g.*, increase the TX power to be higher than the 0dBm used by all experiments in the paper). Third, BLEselect suffers from interference and multi-path effects of different indoor environments, just as other RF systems. We tested the system in a large classroom, a medium living room, and a small and crowded dorm. As expected, the performance of the system deteriorates as the room gets smaller and more crowded. However, we found that hand gestures only have around 5% accuracy decrease at 3m between the classroom and the dorm environment. This shows that some environment-related errors are mutual to both the wrist-worn and the IoT devices, which offsets with each other.

6.1.3 Generalization across Hardware. The hardware manufacturing variances of the RF receiver is inevitable and will lead to AoA estimation performance differences. However, a one-time AoA measurement calibration of the receiver will account for the impact. This can be done by the manufacturer and does not add any burden to users.

6.2 Limitations and Future Work

BLEselect detects gestures and maps them to the target IoT device using AoA data. As a result, our system is not able to differentiate among devices with similar AoA but different distances from the user. RSSI of wireless signals can be used to roughly estimate the distance between the IoT devices and the user. However, the result is usually unreliable due to large RSSI value fluctuations. We plan to explore the possibility of combining AoA and RSSI data so that our system can robustly differentiate devices in the same direction.

Our system suffers from AoA estimation errors due to the small antenna array at the 2.4GHz band and the single radio frond-end of the Bluetooth chip. Even though our system is still able to achieve high selection accuracy, more precise direction-finding will enable exciting applications like object tracking. One way to further reduce AoA errors is to use customized radio hardware at a higher frequency band (*e.g.*, mmWave). In this way, the effective size of the antenna array will be larger, which results in a higher localization resolution. It is even possible to detect the *Encircling* gesture solely based on the periodic signal blockages of the user's arm, so that users do not need to wear a smartwatch.

The current BLEselect system requires users to maintain a straight arm when using hand gestures since it only tracks one wrist-worn device to infer the pointing direction. If a smart ring is worn on the pointing finger in addition to the smartwatch, our system can estimate the pointing direction based on the AoA of both wearable devices. The selection accuracy will also increase if the system applies a user-specific human model to calibrate the pointing direction instead of using one generic model for all users. This will enable more natural and accurate selection using *Pointing* and *Encircling* gestures.

Few false detections of *Nodding* were observed in our user study, which happens when the users were visually searching for the target device. One method to mitigate the false positive issue is to set an absolute threshold for the weighted SVM post-probability so that a device is selected only when it is higher than the threshold and has the largest post-probability. Another method is to collect data and establish a human model for the nodding gesture [45] so that random head movements will not be classified as has a *Nodding* gesture.

7 CONCLUSION

BLEselect estimates the AoA of Bluetooth 5.1 compatible IoT and wrist-worn devices from a pair of smart glasses, then use this information to select the target IoT device. We designed a 5-element 2.4GHz antenna array through simulation that fits the compact size of the smart glasses. We evaluated its AoA estimation performance at an outdoor environment at four different ranges. The results show that the AoA estimation error is less than 10° for 80% of measurements within a 3 meters range for an FOV of $-30° < \theta < 30°, -20° < \phi < 20°$. We then designed three gestures-*Pointing*, *Encircling*, and *Nodding*-for intuitive and natural selection and conducted a user study to determine how to perform each gesture. We also developed a sensing pipeline that supports gesture detection and device selection with one-class SVM models trained in real-time for each hand gesture. Extensive experiments validate our system supports accurate device selection (90.5% averaged across different ranges and separations) with less than 10mW average receiving power and only 1.9mW average power on IoT devices. We also conducted a user study with a diverse set of 22 participants to test our system in more realistic settings. The results show that users can effectively and naturally select IoT devices with an overall accuracy of 96.7%. The results show BLEselect generalize well across users with different ages, heights, and technology savviness. We believe BLEselect has the potential to be integrated into future generations of smart glasses and HMDs in general, which will significantly improve the interaction experience for mixed reality applications.

ACKNOWLEDGMENTS

We thank the reviewers for their helpful feedback. This work is supported by the National Key Research and Development Plan under Grant No.2019YFB1404703 and Young Scientists Fund of the National Natural Science Foundation of China under Grant No.62102401. This work is also supported in part by Academy of Intelligent Computing Technology, Shandong Institutes of Industrial Technology under Grant No.SDAICT2291020, and National Natural Science Foundation of China under Grant No. 62022005, 62272010 and 62061146001.

REFERENCES

[1] 2018. Angle of Arrival BoosterPack. https://dev.ti.com/tirex/explore/node?devtools=BOOSTXL-AOA&node=AHckEvhg0Y3xs5rlangU2w__FUz-xrs__LATEST

[2] 2022. Human head. https://en.wikipedia.org/w/index.php?title=Human_head&oldid=1075721879 Page Version ID: 1075721879.

[3] Abdul Rafey Aftab, Michael von der Beeck, and Michael Feld. 2020. You Have a Point There: Object Selection Inside an Automobile Using Gaze, Head Pose and Finger Pointing. In *Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI '20)*. Association for Computing Machinery, New York, NY, USA, 595–603. https://doi.org/10/gk88qb

[4] Karan Ahuja, Sujeath Pareddy, Robert Xiao, Mayank Goel, and Chris Harrison. 2019. LightAnchors: Appropriating Point Lights for Spatially-Anchored Augmented Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 189–196. https://doi.org/10.1145/3332165.3347884

[5] Ashwin Ashok, Chenren Xu, Tam Vu, Marco Gruteser, Rich Howard, Yanyong Zhang, Narayan Mandayam, Wenjia Yuan, and Kristin Dana. 2016. What Am I Looking At? Low-Power Radio-Optical Beacons for In-View Recognition on Smart-Glass. *IEEE Transactions on Mobile Computing* 15, 12 (Dec. 2016), 3185–3199. https://doi.org/10.1109/TMC.2016.2522967 Conference Name: IEEE Transactions on Mobile Computing.

[6] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: Using the Doppler Effect for Multi-device Interaction. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 583–586. https://doi.org/10/gfsvhm

[7] Zonglong Bai, Liming Shi, Jesper Rindom Jensen, Jinwei Sun, and Mads Græsbøll Christensen. 2021. Acoustic DOA estimation using space alternating sparse Bayesian learning. *EURASIP Journal on Audio, Speech, and Music Processing* 2021, 1 (April 2021), 14. https://doi.org/10.1186/s13636-021-00200-z

[8] Andreas Biri, Neal Jackson, Lothar Thiele, Pat Pannuto, and Prabal Dutta. 2020. SociTrack: infrastructure-free interaction tracking through mobile sensor networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. ACM, London United Kingdom, 1–14. https://doi.org/10.1145/3372224.3419190

[9] Leo Botler, Michael Spörk, Konrad Diwold, and Kay Römer. 2020. Direction Finding with UWB and BLE: A Comparative Study. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 44–52. https://doi.org/10.1109/MASS50613.2020.00016 ISSN: 2155-6814.

[10] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (Jan. 2018), 129:1–129:27. https://doi.org/10/gfhwdh

[11] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. 2017. RF-Echo: A Non-Line-of-Sight Indoor Localization System Using a Low-Power Active RF Reflector ASIC Tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 222–234. https://doi.org/10.1145/3117811.3117840

[12] Marco Cominelli, Paul Patras, and Francesco Gringoli. 2019. Dead on Arrival: An Empirical Study of The Bluetooth 5.1 Positioning System. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH '19)*. Association for Computing Machinery, New York, NY, USA, 13–20. https://doi.org/10/gg9m69

[13] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10/gg4nkk

[14] Ziheng Ding, Jingfeng Chen, Chong He, and Ronghong Jin. 2022. Elevation and Azimuth Direction Finding by Two-Element Pattern Reconfigurable Antenna Array. *IEEE Transactions on Antennas and Propagation* 70, 3 (March 2022), 2261–2270. https://doi.org/10.1109/TAP.2021.3118820 Conference Name: IEEE Transactions on Antennas and Propagation.

[15] Egils Ginters and Jorge Martin-Gutierrez. 2013. Low Cost Augmented Reality and RFID Application for Logistics Items Visualization. *Procedia Computer Science* 26 (Jan. 2013), 3–13. https://doi.org/10.1016/j.procs.2013.12.002

[16] Cory Hekimian-Williams, Brandon Grant, Xiuwen Liu, Zhenghao Zhang, and Piyush Kumar. 2010. Accurate localization of RFID tags using phase difference. In *2010 IEEE International Conference on RFID (IEEE RFID 2010)*. 89–96. https://doi.org/10.1109/RFID.2010.5467268 ISSN: 2374-0221.

[17] Jie Hua, Sangsu Lee, Gruia-Catalin Roman, and Christine Julien. 2021. ArcIoT: Enabling Intuitive Device Control in the Internet of Things through Augmented Reality. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 558–564. https://doi.org/10/gpnd3w

[18] Dongsik Jo and Gerard Jounghyun Kim. 2016. ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere. *IEEE Transactions on Consumer Electronics* 62, 3 (Aug. 2016), 334–340. https://doi.org/10.1109/TCE.2016.7613201 Conference Name: IEEE Transactions on Consumer Electronics.

[19] Kyu-Han Kim. 2019. When IoT met Augmented Reality: Visualizing the Source of the Wireless Signal in AR View. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, Seoul Republic of Korea, 117–129. https://doi.org/10/gkfcrs

[20] Quan Kong, Takuya Maekawa, Taiki Miyanishi, and Takayuki Suyama. 2016. Selecting Home Appliances with Smart Glass Based on Contextual Information. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 97–108. https://doi.org/10/gnc4j4

[21] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. *ACM SIGCOMM Computer Communication Review* 45, 4 (Aug. 2015), 269–282. https://doi.org/10.1145/2829988.2787487

[22] Hanchuan Li, Eric Whitmire, Alex Mariakakis, Victor Chan, Alanson P. Sample, and Shwetak N. Patel. 2019. IDCam: Precise Item Identification for AR Enhanced Object Interactions. In *2019 IEEE International Conference on RFID (RFID)*. 1–7. https://doi.org/10.1109/RFID.2019.8719279 ISSN: 2573-7635.

[23] Robert LiKamWa, Zhen Wang, Aaron Carroll, Felix Xiaozhu Lin, and Lin Zhong. 2014. Draining our glass: an energy and heat characterization of Google Glass. In *Proceedings of 5th Asia-Pacific Workshop on Systems (APSys '14)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/2637166.2637230

[24] Sikun Lin, Hao Fei Cheng, Weikai Li, Zhanpeng Huang, Pan Hui, and Christoph Peylo. 2017. Ubii: Physical World Interaction Through Augmented Reality. *IEEE Transactions on Mobile Computing* 16, 3 (March 2017), 872–885. https://doi.org/10.1109/tmc.2016.2567378 Conference Name: IEEE Transactions on Mobile Computing.

[25] Xuefeng Liu, Tianye Yang, Shaojie Tang, Peng Guo, and Jianwei Niu. 2020. From relative azimuth to absolute location: pushing the limit of PIR sensor based localization. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3372224.3380878

[26] R. Mayrhofer and H. Gellersen. 2009. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing* 8, 6 (June 2009), 792–806. https://doi.org/10/bqk9kd

[27] Georg Oberholzer, Philipp Sommer, and Roger Wattenhofer. 2011. SpiderBat: Augmenting wireless sensor networks with distance and angle information. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. 211–222.

[28] Michael Parker. 2010. Chapter 18 - Synthetic Array Radar. In *Digital Signal Processing 101*, Michael Parker (Ed.). Newnes, Boston, 213–222. https://doi.org/10.1016/B978-1-85617-921-8.00022-5

[29] Shwetak N. Patel and Gregory D. Abowd. 2003. A 2-Way Laser-Assisted Selection Scheme for Handhelds in a Physical Environment. In *UbiComp 2003: Ubiquitous Computing (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 200–207. https://doi.org/10.1007/978-3-540-39653-6_16

[30] Giovanni Pau, Fabio Arena, Yonas Engida Gebremariam, and Ilsun You. 2021. Bluetooth 5.1: An Analysis of Direction Finding Capability for High-Precision Location Services. *Sensors* 21, 11 (May 2021), 3589. https://doi.org/10.3390/s21113589

[31] Chunyi Peng, Guobin Shen, Yongguang Zhang, and Songwu Lu. 2009. Point&Connect: Intention-based Device Pairing for Mobile Phone Users. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. ACM, New York, NY, USA, 137–150. https://doi.org/10/b6rcdg

[32] Felix Putze, Dennis Weiß, Lisa-Marie Vortmann, and Tanja Schultz. 2019. Augmented Reality Interface for Smart Home Control using SSVEP-BCI and Eye Gaze. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 2812–2817. https://doi.org/10.1109/SMC.2019.8914390 ISSN: 2577-1655.

[33] Jonathan Rosales, Sourabh Deshpande, and Sam Anand. 2021. IIoT based Augmented Reality for Factory Data Collection and Visualization. *Procedia Manufacturing* 53 (Jan. 2021), 618–627. https://doi.org/10.1016/j.promfg.2021.06.062

[34] Ludwig Sidenmark and Hans Gellersen. 2019. Eye&Head: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1161–1174. https://doi.org/10.1145/3332165.3347921

[35] Elahe Soltanaghaei, Adwait Dongare, Akarsh Prabhakara, Swarun Kumar, Anthony Rowe, and Kamin Whitehouse. 2021. TagFi: Locating Ultra-Low Power WiFi Tags Using Unmodified WiFi Infrastructure. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (March 2021), 1–29. https://doi.org/10/gjnjgp

[36] Elahe Soltanaghaei, Akarsh Prabhakara, Artur Balanuta, Matthew Anderson, Jan M. Rabaey, Swarun Kumar, and Anthony Rowe. 2021. Millimetro: mmWave retro-reflective tags for accurate, long range localization. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. ACM, New Orleans Louisiana, 69–82. https://doi.org/10/gjg3rb

[37] Zheng Sun, Aveek Purohit, Raja Bose, and Pei Zhang. 2013. Spartacus: Spatially-aware Interaction for Mobile Devices Through Energy-efficient Audio Sensing. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA, 263–276. https://doi.org/10/gk5cwm

[38] David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017. Smart Home Control Using Motion Matching and Smart Watches. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 466–468. https://doi.org/10/gk5cwj

[39] Jue Wang and Dina Katabi. 2013. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 51–62. https://doi.org/10.1145/2486001.2486029

[40] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: virtual touch screen in the air using RF signals. In *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, Chicago Illinois USA, 235–246. https://doi.org/10.1145/2619239.2626330

[41] Weiguo Wang, Jinming Li, Yuan He, and Yunhao Liu. 2020. Symphony: localizing multiple acoustic sources with a single microphone array. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. Association for Computing Machinery, New York, NY, USA, 82–94. https://doi.org/10.1145/3384419.3430724

[42] Yuntao Wang, Jiexin Ding, Ishan Chatterjee, Farshid Salemi Parizi, Yuzhou Zhuang, Yukang Yan, Shwetak Patel, and Yuanchun Shi. 2022. FaceOri: Tracking Head Position and Orientation Using Ultrasonic Ranging on Earphones. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3491102.3517698

[43] Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4000–4008. https://doi.org/10/gg94n6

[44] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *10th ${$USENIX$}$ Symposium on Networked Systems Design and Implementation (${$NSDI$}$ 13)*. 71–84.

[45] Yukang Yan, Yingtian Shi, Chun Yu, and Yuanchun Shi. 2020. HeadCross: Exploring Head-Based Crossing Selection on Head-Mounted Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (March 2020), 35:1–35:22. https://doi.org/10.1145/3380983

[46] Jackie (Junrui) Yang and James A. Landay. 2019. InfoLED: Augmenting LED Indicator Lights for Device Positioning and Communication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 175–187. https://doi.org/10.1145/3332165.3347954

[47] Lichen Yao. 2018. Bluetooth Direction Finding. (2018).

[48] Ben Zhang, Yu-Hsiang Chen, Claire Tuna, Achal Dave, Yang Li, Edward Lee, and Björn Hartmann. 2014. HOBS: Head Orientation-based Selection in Physical Spaces. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction (SUI '14)*. ACM, New York, NY, USA, 17–25. https://doi.org/10/gk5cwk

[49] Tengxiang Zhang, Xin Yi, Ruolin Wang, Jiayuan Gao, Yuntao Wang, Chun Yu, Simin Li, and Yuanchun Shi. 2019. Facilitating Temporal Synchronous Target Selection through User Behavior Modeling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (Dec. 2019), 1–24. https://doi.org/10/gpnbc5

[50] Tengxiang Zhang, Xin Yi, Ruolin Wang, Yuntao Wang, Chun Yu, Yiqin Lu, and Yuanchun Shi. 2018. Tap-to-Pair: Associating Wireless Devices with Synchronous Tapping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4 (Dec. 2018), 201:1–201:21. https://doi.org/10/gg7k4k

[51] Tengxiang Zhang, Xin Zeng, Yinshuai Zhang, Ke Sun, Yuntao Wang, and Yiqiang Chen. 2020. ThermalRing: Gesture and Tag Inputs Enabled by a Thermal Imaging Smart Ring. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, Honolulu, HI, USA, 1–13. https://doi.org/10.1145/3313831.3376323 00000.

A 2D MUSIC AOA ESTIMATION ALGORITHM

The wireless signal can be modeled as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), t = 1, 2, ..., N, \tag{7}$$

where $\mathbf{x}(t)$ $\mathbf{A}$, $\mathbf{s}(t)$ and $\mathbf{n}(t)$ denote the received signal vector, the ideal steering matrix, the source signal vector and the Gaussian noise term with σ_N^2 variance.

$\mathbf{x}(t)$ is defined by the output of the array with phase $\alpha_i(t)$ in each antenna generated using the IQ value in the CTE packet.

$$\mathbf{x}(t) = \begin{bmatrix} \cos\alpha_1(t) + j\sin\alpha_1(t) \\ \cos\alpha_2(t) + j\sin\alpha_2(t) \\ \cos\alpha_3(t) + j\sin\alpha_4(t) \\ \cos\alpha_4(t) + j\sin\alpha_4(t) \\ \cos\alpha_5(t) + j\sin\alpha_5(t) \end{bmatrix} \tag{8}$$

and $\mathbf{A}$ is the steering vector with d_{x_i}, d_{y_i}, β_1 , β_2 to be the position of each antenna on the glass in the XY plane, the projected direction in the XY plane and direction angle with Z axis depicted in the Figure5.

$$\mathbf{A} = a(\beta_1, \beta_2) = \begin{bmatrix} e^{j\frac{2\pi}{\lambda}(d_{x_1}\sin\beta_2\cos\beta_1 + d_{y_1}\sin\beta_2\sin\beta_1)} \\ e^{j\frac{2\pi}{\lambda}(d_{x_2}\sin\beta_2\cos\beta_1 + d_{y_2}\sin\beta_2\sin\beta_1)} \\ e^{j\frac{2\pi}{\lambda}(d_{x_3}\sin\beta_2\cos\beta_1 + d_{y_3}\sin\beta_2\sin\beta_1)} \\ e^{j\frac{2\pi}{\lambda}(d_{x_4}\sin\beta_2\cos\beta_1 + d_{y_4}\sin\beta_2\sin\beta_1)} \\ e^{j\frac{2\pi}{\lambda}(d_{x_5}\sin\beta_2\cos\beta_1 + d_{y_5}\sin\beta_2\sin\beta_1)} \end{bmatrix} \tag{9}$$

In actual practice, the covariance matrix of the received signal is roughly estimated by the time average.

$$R_{xx} \approx \frac{1}{N}\sum_{t=1}^{N} x(t)x^H(t). \tag{10}$$

MUSIC then uses the eigenvectors decomposition and eigenvalues of the covariance matrix to separate the signal space and noise space . The formula is

$$R_{xx} = V\Lambda V^{-1}, \tag{11}$$

where Λ and V consist of eigenvalues and corresponding eigenvectors.

The eigenvectors are sorted by the value of corresponding eigenvalues. Consider the first eigenvector with the biggest eigenvalue as the signal space and the remaining four eigenvalues and eigenvectors as the noise space. Then we obtain the noise vector $\mathbf{V}_n = [v_2, v_3, v_4, v_5]$. Finally, we can get a pseudo spectrum according to the following formula:

$$P(\beta_1, \beta_2) = \frac{1}{\mathbf{a}(\beta_1, \beta_2)^H \mathbf{V}_n \mathbf{V}_n^H \mathbf{a}(\beta_1, \beta_2)}. \tag{12}$$

By selecting the spectrum's peak, we obtain the direction of the transmitted signal in the 3D space with direction β_1 and β_2. Furthermore, we projected the direction vector onto the YZ plane and obtained the truth azimuth angle ϕ and elevation angle θ of the device according to the following equations

$$\begin{aligned} \phi &= \arctan\left(\frac{\cot\beta_2}{\cos\beta_1}\right) \\ \theta &= \arctan\left(\frac{\sin\beta_1}{\cos^2\beta_1 + \cot^2\beta_2}\right) \end{aligned} \tag{13}$$

B SYSTEM PARAMETERS

We mark the system parameters in red in Figure 3. We summarize their names, functions, values, setting rationals, and generalizability below.

c_1: RF Receiver Calibration Parameter The RF receiver circuits and the antenna array of BLEselect suffer from system errors and manufacture variances. Even though such errors will be offset for hand gestures, a one-time calibration is necessary for *Nodding*. We measured our prototype's angle offset and used a linear compensation model ax+b for correction. The calibration is hardware-specific and generalizes to all users. We scale the phase value of each antenna to $[0.8x - 4.7\ 0\ 1.2x - 19.9\ 0.9x - 13.5\ x + 9.8]$ in the same order of Figure 5a.

c_2: Pointing Direction Estimation Parameter In this paper, the system uses human body data measured from a 178cm male to transform the AoA of a wrist-worn device to the pointing direction (details explained in Section 3.2.2). Different users may have different body structures, which leads to estimation errors. However, we measured body data of 6 people with vastly different heights (156cm-184cm) and found that the maximal 2D angle error is only 0.48°(Figure 5c). The user study results also validate that the impact of body heights on the selection accuracy is minimal.

w: Signal Processing Window Size and Step The system uses a window size of 0.8s and a step size of 0.08s to detect gestures based on the estimated AoA data. Such a window size increases the chances to capture the gestures. The small step reduces the possibilities of false negatives.

TH_1: Nodding Detection Angle Threshold The system analyzes the angle changes inside a window of the device that is closed to the center. When the standard deviations of the azimuth and elevation angles of the device satisfy $\sigma_{az.}$ <4°,$\sigma_{el.}$ >28°, a *Nodding* gesture is detected. The values of $TH_1 - TH_4$ and other relevant parameters are all determined through a pilot study with 4 participants. Each was asked to nod, raise/drop hand, point, and draw an in-air circle naturally for 30 times.

TH_2: Hand Gesture Segmentation RSSI Threshold A hand gesturing period starts with a hand-up event and stops at a hand-down event. A hand-up event is detected if the RSSI of the wrist-worn device rises above $-44dBm$ for 0.4s, and a hand-down event is detected when the RSSI falls below $-44dB$ for 0.4s. The threshold can be different for different wrist-worn devices (*e.g.*, with different TX power or antenna gain).

TH_3: Pointing Detection Angle Threshold The system detects *Pointing* if the AoA standard deviation of the wrist-worn device is smaller than 3°within the hand gesturing period. A larger threshold can tolerate hand movements during pointing, but will increase false positives.

TH_4: Encircling Detection Frequency Threshold The system first performs a 64-point FFT on the AoA data of the wrist-worn device within the hand gesturing period. *Encircling* is detected if the frequency component for 0.5Hz is larger than 4.5.

V_{data}: Training Data Generation Parameters We generate training data that mimics the possible AoA of the target device during a selection gesture. For *Nodding*, we conducted a pilot study with 4 participants nodding at an IoT device. The result showed that its AoA data falls into the area $-5 < \theta < 5, -10 < \phi < 30$. We then generate training data for the *Nodding* model by sampling the area with a 0.5°step in both directions.
For *Encircling*, such an area is generated in real-time after each gesturing period, which is enclosed by the trajectory of the pointing direction. We use a Kalman filter ($Q = 10^{-4}, R = 4 \times 10^{-5}$) to smooth the trajectory. Similarly, we generate the training data for the *Encircling* model by sampling the area with a 0.5°step size in both directions. For *Pointing*, on the other hand, no new data is generated. We simply use the pointing direction data during the gesturing period to train the model in real-time.