

GWrite: Enabling Through-the-Wall Gesture Writing Recognition Using WiFi

Sai Deepika Regani, Beibei Wang, Yuqian Hu, and K. J. Ray Liu

Abstract—Recognizing in-air gestures can enable intelligent Human-Computer Interaction (HCI) applications and facilitate human lives. However, existing sensor/camera-based methods for gesture recognition are either non-ubiquitous, intrusive to privacy, or inconvenient to carry around. Contemporary device-free approaches require the person to be in the line of sight and proximity to the sensing device. This paper shows that WiFi signals can recognize hand-drawn in-air gestures even when the gesture location is non-line-of-sight/beyond walls to the WiFi transceivers. The proposed *GWrite* system utilizes the CSI time-series information from commercial WiFi chipsets. *GWrite* employs a unique approach for performing hand gestures, thus enabling the design of a hand movement model. Using the model and the time-reversal (TR) technique, this work derives a correspondence between the similarity of CSIs and the relative distance moved by the hand. This relation gave rise to unique features such as the number of segments, angle, and the intersection between segments that can classify a set of gesture shapes consisting of straight-line segments. *GWrite* achieved an accuracy of 92% on a group of 15 gestures. The proposed approach can be applied to a broader set of gestures, unlike the current systems that function over a limited gesture set.

Index Terms—Gesture Model, Channel State Information (CSI), Time Reversal Resonating Strength (TRRS), WiFi Sensing, HCI.

I. INTRODUCTION

A proliferating number of intelligent connected devices and computers are transforming the space around us. Human-Computer Interaction (HCI) forms a significant portion of our daily routine, and there is a need to simplify it as much as possible. With time, researchers worldwide have come up with innovative solutions and approaches to ease HCI. We have come a long way from manual operations to push buttons, touch sensors to motion sensors, and the fancier fingerprint and face ID systems for authenticated services. We evolved from using keyboard inputs to using in-the-air gestures and voice commands to talk to computers. These efforts underscore our desire to ease interaction with computers and facilitate human lives.

In-the-air gesture recognition blends easily with future smart environments. Researchers have experimented with different signals to build practical gesture recognition systems. Vision-based approaches tracked the movement of the hand/finger to reconstruct the hand trajectory. Commercial applications such as gaming have become widely popular with the Microsoft Kinect and LeapMotion. The vision-based approaches

D. Regani, B. Wang, Y. Hu, and K. J. R. Liu are with Origin Wireless, Inc., Greenbelt, MD 20770 USA. (e-mail: {rdeepika, yhu1109}@terpmail.umd.edu; {beibei.wang, ray.liu}@originwirelessai.com).

specifically target LOS applications, require ambient light, and potentially raise privacy concerns. Sensor-based techniques provide reliable solutions to gesture recognition, and many such systems have been developed [1], [2]. However, passive or device-free approaches are preferred due to better user experience. A few works used RF signals to track hand motion and perform handwriting/gesture recognition [1], which usually require dedicated hardware and incur additional costs. On the other hand, passive WiFi-based approaches have the most suitable features, such as easy penetration of walls relaxing the LOS constraints, ubiquitous infrastructure, privacy preservation, and better user experience, for many exciting applications, including practical gesture recognition [3]–[7]. In this work, we propose *GWrite*, a device-free gesture writing recognition system using commodity WiFi devices that can work in through-the-wall scenarios. Building such a system entails multiple challenges.

Firstly, given the low bandwidths in commercial WiFi systems (20-80 MHz), it is difficult to isolate the signal of interest, i.e., corresponding to the moving hand. For example, with a 40 MHz bandwidth, all the multipath with a path length differing by 7.5 m are superimposed on a single Channel Impulse Response (CIR) tap. Due to this, past works directly matched the CSI time series for a set of pre-defined gestures and locations between the training and testing phases. Such approaches had several drawbacks, which were later pointed out in the literature by the terms “location/user dependency” and “pattern inconsistency.” Difficulty in isolating the signal in time also forbids using features such as Doppler velocity or time of flight (ToF) to determine the speed/distance moved by the target. Further, the spatial information cannot be extracted due to the NLOS setup and the absence of antenna arrays. With *GWrite*, we propose a workaround to overcome the above constraints. Instead of directly matching the CSIs in the online and offline phases, we develop a correspondence between the relative distance moved by the hand and the drop in the similarity value within the CSI time series. Although the absolute direction/orientation information is absent, we perform the gesture classification using features that describe the relative shape of the gesture trajectory.

Secondly, the commercial WiFi lies either in the 2.4 GHz or 5 GHz bands in which the electromagnetic (EM) waves can easily penetrate walls and span typical indoor spaces. Also, they are not easily attenuated with distance. While these features make it an ideal candidate for through-the-wall applications, the signal reflected off the hand is completely immersed and causes a minimal perturbation in the intense multipath of an indoor environment. This behavior has forced

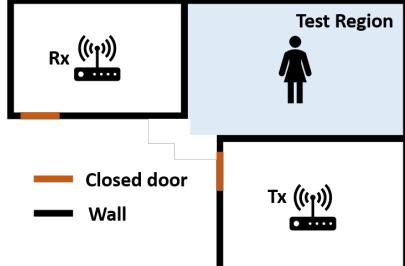


Fig. 1: A typical through-the-wall experimental setup.

most previous works to limit their coverage to Line-of-Sight (LOS) or distances around 1m allowing the signal of interest to dominate other irrelevant signals from the static environment. We address this problem by removing the contribution from the static environment/scatterers and using Maximal-Ratio Combining (MRC) to boost the dynamic signal generated by the hand movement and suppress noise. A typical through-the-wall experimental setup is as shown in Fig. 1.

Lastly, different parts of the hand move with different velocities during a hand gesture and have comparable contributions to the channel perturbation. Building a physical model, therefore, becomes challenging for hand gestures. In the past, researchers have modeled either whole-body movements focusing on the speed of the torso [8] or used finger gestures that can be more controlled [9]. Differently, in this work, we propose a hand gesture model by specifying a unique method of performing gestures and deriving a mathematical expression relating the degree of hand movement and the similarity decay between the CSIs.

At the core of the *GWrite* system lies a statistical model of the WiFi channel time-variance. The model allowed us to simplify the expression for the adopted similarity metric between two CIRs. The proposed unique hand gestures further helped us derive a correspondence between the relative distance moved by the hand and the similarity decay. Leveraging this property, we extracted differentiable features for gesture shapes consisting of straight-line segments. We designed algorithms to automatically identify straight line segments, determine the angle between two segments and identify intersections between segments. The likelihood for each gesture shape is calculated by accumulating the probability scores for each feature. The gesture shape is identified as the one with the highest likelihood score. The contributions of this work can be summarized as follows:

- We proposed *GWrite*, a gesture writing recognition system that can work in a through-the-wall setting and recognize gesture shapes composed of straight-line segments. We have developed a complete pipeline of such a system, including gesture segmentation, turn angle classification, and intersection detection.
- We proposed a unique model for hand movement during a gesture. Leveraging the model, we derived a relation between the perturbations in the physical environment and the similarity within the CSI time series in a through-the-wall setting. We believe such an understanding will

open diverse opportunities to develop practical gesture recognition systems and expand the existing WiFi-based systems restricted only to LOS setups today.

- We built a prototype of the proposed system using COTS WiFi devices and evaluated the performance using a set of 15 uppercase English characters. The classification accuracy achieved was 92% which can be further improved as more bandwidth becomes available.

The rest of the paper is organized as follows. Section II summarizes the previous works on gesture recognition. Section III introduces the proposed gesture model and the concept of TRRS decay. Section IV describes the algorithm to extract the features of the gesture trajectories and classify gestures. The performance evaluation of *GWrite* is presented in Section V. Micro-benchmarks and the influence of various factors are discussed in Section VI and finally, Section VIII concludes the paper.

II. RELATED WORKS

Research on passive gesture recognition has come a long way. It was observed that any physical movement/actions in the environment caused perturbation to the wireless channel and the associated multipath profile. Researchers started to analyze these changes and relate them with the respective activities to design useful applications. Initial efforts in this direction used the Received Signal Strength Indicator (RSSI) to realize different applications including gesture recognition, target tracking and breathing estimation [10]–[13]. However, RSSI, a single scalar, could only provide coarse information about the activities leading to the channel changes and was also sensitive to environmental changes. With the release of the Channel State Information (CSI) tool for commercial WiFi devices [14], fine-grained information on the channel measurement became available, opening up many opportunities to achieve new applications and improve the systems which, till then, depended on RSSI.

A given movement/activity results in a repeatable CSI time series when the physical environment is relatively unchanged. Thus, many works maintained databases of CSI patterns in the training phase and compared them with the CSIs in the testing phase using a similarity metric such as Dynamic Time Warping (DTW) or by using neural networks [15]. Although such an approach should work in theory, it is practically impossible to build a training database of CSI prototypes spanning a wide range of possible locations, users, and environments. Further, the application was restricted to a limited set of pre-defined gestures with available CSI patterns. A few works tried to overcome this problem by post-processing the CSI using different filtering and feature selection techniques such as PCA, subcarrier selection, wavelet transforms, and low-pass filtering [16]–[20]. However, these attempts were not successful in obtaining location-independent features characteristic of movement patterns.

Features such as the direction of gesture [21] and the number of repetitions [22] were later used, which are location independent. Such approaches also worked for the Non-Line-of-Sight (NLOS) and through-the-wall setups, thereby relaxing

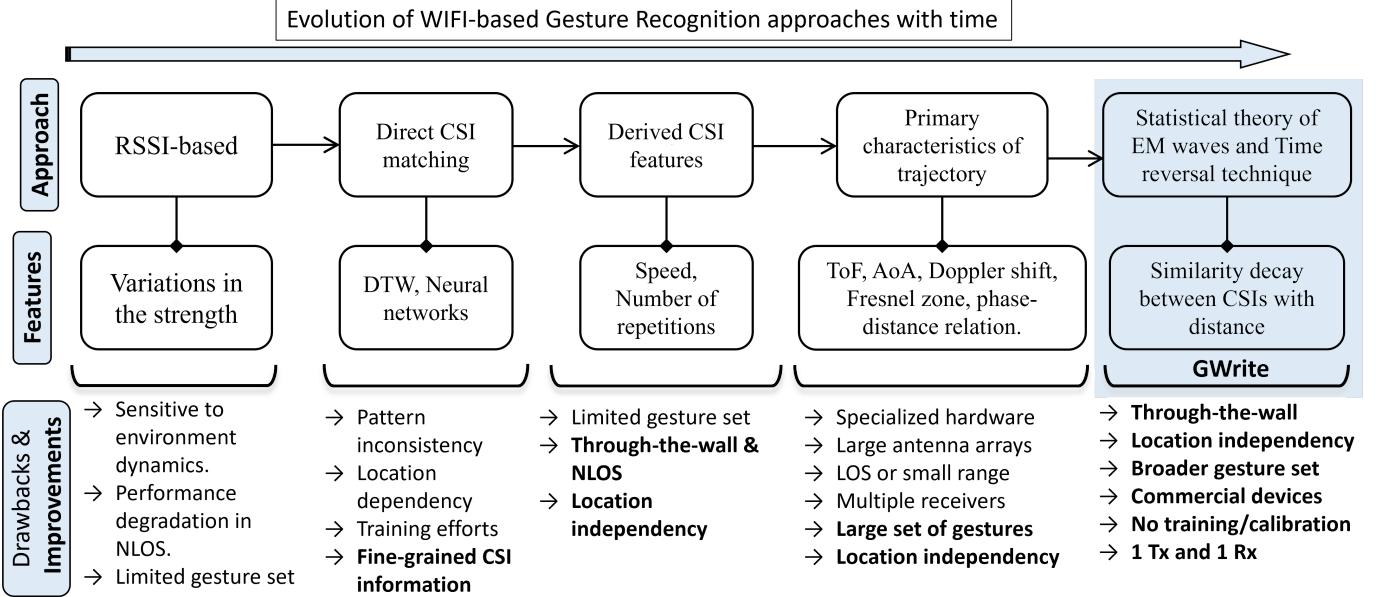


Fig. 2: Evolution of the WIFI-based gesture recognition approaches. The improvements in each stage motivating the current through-the-wall robust gesture recognition are highlighted in bold text

the constraints on the placement of devices. However, with such features, the number of differentiable gestures is limited. Given the number of smart devices surrounding us today, a broader set of gestures is required for interaction and to add more flexibility to the user. The intuitive way forward to achieve a more comprehensive set of recognizable gestures is to design features that describe the original trajectory of the hand motion or, in other words, that can reconstruct it.

The relative location of the hand with time has to be known to reconstruct the gesture trajectory. For this, information such as Time-of-Flight to determine the distance/range of the hand from the receiver, Angle-of-Arrival (AOA) [23], [24] to determine the angular location, and Doppler shift to determine the radial velocity of the hand [25] were used. However, all these approaches require modified hardware or a strict LOS set up with a small operational range. *WiDraw* relied on the signal strength from multiple APs to determine the location of the hand at different time instances and construct the trajectory [24]. A centimeter accuracy finger tracking was achieved using two receivers placed orthogonally in FingerDraw [9]. These systems are effective only up to a small range (<1 m). As the distance between the hand and the transceivers increases, isolating the signal reflected off the hand becomes more challenging.

With *GWrite*, we propose a through-the-wall gesture writing recognition system that can extract information about the gesture trajectory, thus facilitating a large set of recognizable gestures. Furthermore, the time-reversal (TR) technique, combined with the unique gesture model, enabled us to extract features that reconstruct the gesture shape. A summary of how the WiFi-based gesture recognition approaches have evolved with time is shown in Fig 2.

III. GWRITE GESTURE MODEL

In this section, we develop a statistical model to explain the changes in the wireless channel induced by a hand gesture. An instance of a wireless channel can be characterized by its Channel State Information (CSI) or equivalently by its Channel Impulse Response (CIR). A gesture is a continuous movement of the hand and can thus be represented by a CIR/CSI time series. It is useful to quantify the relationships/similarities between any two CSIs in the time series to develop a correspondence between the hand movement and the wireless channel variations. For this, we adopt the Time Reversal Resonating Strength (TRRS) metric [26].

Assume the wireless channel between a transmitter (Tx) and a receiver (Rx) is denoted by the CIR $h(l)$, where $l = 0, 1, \dots, L - 1$ and L is the total number of taps. Also assume that the CIR is normalized i.e., $\sum_{l=0}^{L-1} |h(l)|^2 = 1$. If the Tx sends an impulse $\delta(l)$, the Rx receives $y(l) = h(l)$. Note that the wireless channel from Tx→Rx and Rx→Tx can be considered equivalent following the channel reciprocity. Now, if the Rx sends $h^*(-l)$, i.e., a time reversed and conjugated version of the channel, then the received signal at the Tx becomes $y'(l) = h^*(-l) \star h(l)$, where \star denotes convolution. The zeroth tap of $y'(l)$ becomes,

$$y'(0) = \sum_{l=0}^{L-1} h^*(l)h(l) = \sum_{l=0}^{L-1} |h(l)|^2 = 1, \quad (1)$$

i.e., we can know if the channel at a later time instance is different from the initial channel by sending a time reversed and conjugate version of the initial CIR from Rx and recording $y'(0)$ at Tx. It can be concluded that the channel has changed

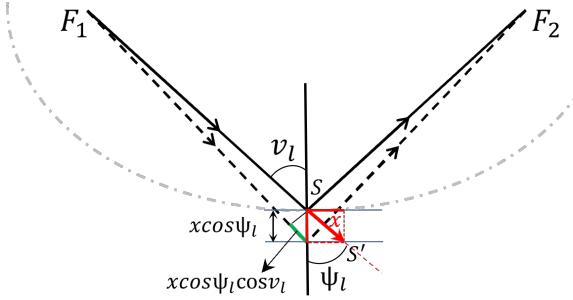


Fig. 3: Demonstration of the multipath length difference when a reflector moves by a small distance x .

if $y'(0) \neq 1$. Alternatively, we can compute,

$$\eta(\mathbf{h}_1, \mathbf{h}_2) = \left| \frac{\sum_{l=0}^{L-1} h_1(l) h_2(l)^*}{\sqrt{\sum_{l=0}^{L-1} |h_1(l)|^2} \sqrt{\sum_{l=0}^{L-1} |h_2(l)|^2}} \right|, \quad (2)$$

which gives the TRRS between two CIRs \mathbf{h}_1 and \mathbf{h}_2 . The value of η lies between 0 and 1 where 0 is achieved by a pair of uncorrelated CIRs. Assessing the extent of the physical perturbation in the channel could reveal more information about the hand movement than mere detection of a change. Let us further analyze (2) for the case of a hand gesture to understand the relation between the TRRS drop and the extent of hand movement.

Let \mathbf{h}_0 denote the CIR when the hand is at an initial position. Due to the multiple reflections in an indoor environment, each tap of the CIR is composed of several multipath components (MPCs) which can be explicitly written as:

$$h_0[l] = \sum_{m \in M} \zeta_{l,m} e^{-j2\pi f_c \tau_{0,l}(m)}, \quad (3)$$

where M is the set of multipath, ζ is the complex path gain, f_c is the carrier frequency and τ is the path delay. Let \mathbf{h}_x denote the CIR after the hand moved by a short distance x . Then,

$$h_0[l] h_x[l]^* = \sum_{m \in M} \zeta_{l,m} e^{-j2\pi f_c \tau_{0,l}(m)} \sum_{m \in M} \zeta_{l,m}^* e^{j2\pi f_c \tau_{x,l}(m)}, \quad (4)$$

where the set M is considered unchanged during the short time. Assuming a sufficiently large bandwidth such that the significant MPCs are captured on distinct CIR taps [26], we can approximate the numerator of $\eta(\mathbf{h}_0, \mathbf{h}_x)$ as:

$$\left| \sum_{l=0}^{L-1} h_0[l] h_x[l]^* \right| \approx \left| \sum_{l=0}^{L-1} |\zeta_l|^2 e^{j2\pi f_c (\tau_{x,l} - \tau_{0,l})} \right|. \quad (5)$$

Eq. (5) does not indicate a direct relationship between the distance moved (x) and the drop/decay in the TRRS value. To make this correspondence more evident, we simplify the equation by exploiting the following observations.

Path length difference: The change in the path length when the hand moves by a small distance x can be calculated using the angle of reflection v_l and the angle between the surface normal and the moving direction, ψ_l as indicated in Fig. 3. F_1 and F_2 are the Tx and Rx locations, which can be considered an ellipse's foci. By the geometric definition of an ellipse,

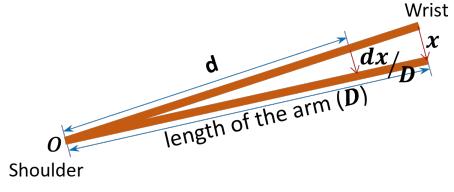


Fig. 4: Demonstration of the movement of the hand pivoted at the shoulder (O).

a small displacement along the surface does not impact the total length of the path. Therefore, the change in the total path length depends only on the displacement along the normal to the reflecting surface/ellipse. The difference in the delays of a path l when the hand moves a distance x can be written as,

$$\tau_{x,l} - \tau_{0,l} = \frac{2x \cos \psi_l \cos v_l}{c}. \quad (6)$$

Uniform scattering: In a rich indoor environment with a lot of multipath, we can assume that the MPCs are uniformly distributed in the space. This allows us to model the angle of incidence of the MPCs at any point in space to be uniformly distributed in $(0, 2\pi]$, converting the summation in (5) to an integration over the angles as,

$$\sum_{l=0}^{L-1} h_0[l] h_x[l]^* \approx K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} e^{(j2\pi f_c \frac{2x \cos \psi_l \cos v_l}{c})} dv_l d\psi_l, \quad (7)$$

where K is the normalization constant.

Fraction of multipaths: Amongst the dense multipath, only a fraction of the MPCs are influenced by the hand's movement. If we denote the fraction of the multipath contributed by reflections from the static environment by T_S , the RHS of (7) can be written as,

$$K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} T_S + (1 - T_S) e^{j2\pi f_c \frac{2x \cos \psi_l \cos v_l}{c}} dv_l d\psi_l \quad (8)$$

$$= KT_S(2\pi)^2 + K(1 - T_S)(2\pi)^2 J_o^2 \left(\frac{2\pi f_c x}{c} \right), \quad (9)$$

where $J_0(\cdot)$ is a zeroth-order Bessel function of the first kind [27].

Stretched hand gesture: During a hand gesture, different parts of the hand move at different speeds and in different directions. Modeling an arbitrary hand movement is, therefore, complicated. In this work, we propose a unique way of performing a gesture with a hand, as shown in Fig. 4 with the arm stretched and pivoted about the shoulder O . We will also show the performance of *GWrite* when relaxing this constraint in Section VI-E. The implications of such a constrained movement are three-fold.

- The spatial orientation of all the parts of the arm can be determined from the wrist's location relative to the shoulder.
- The locus of the moving scatterers (different parts of the hand) is unique and does not overlap/intersect with each other. In a gesture, the only way for a part of the arm to repeat a spatial location is when the whole arm repeats its location.

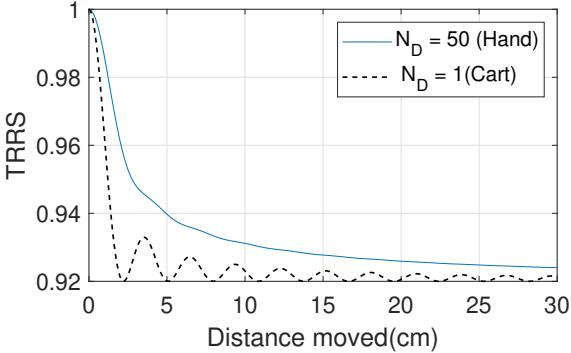


Fig. 5: TRRS decay in case of hand movement (dynamic scatterers with linearly increasing speed) and a cart movement (dynamic scatterers with one speed).

- When the hand is moved per Fig. 4, the hand can be modeled as a combination of multiple scatterers whose displacement increases linearly from 0 to x , where x is the displacement of the wrist. If the wrist is displaced by x , then a point on the arm at a distance d from the shoulder is displaced by $\frac{dx}{D}$, where D is the total length of the arm. The dynamic multipath portion of (9) can be further split into the contribution of N_D scatterers along the arm's length. This results in a summation in the second term of (9), whose RHS now becomes,

$$KT_S(2\pi)^2 + \frac{K(1-T_S)(2\pi)^2}{N_D} \sum_{i=1}^{N_D} J_o^2 \left(\frac{2\pi f_c x i}{c N_D} \right), \quad (10)$$

where N_D is the total number of dynamic scatterers that together form the moving hand.

The denominator of (2) becomes,

$$\sqrt{\sum_{l=0}^{L-1} |h_1[l]|^2} \sqrt{\sum_{l=0}^{L-1} |h_2[l]|^2} = K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} 1 d\nu_l d\psi_l. \quad (11)$$

Using (10) and (11), the TRRS between \mathbf{h}_0 and \mathbf{h}_x can be written as,

$$\eta(\mathbf{h}_0, \mathbf{h}_x) = T_S + \frac{(1-T_S)}{N_D} \sum_{i=1}^{N_D} J_o^2 \left(\frac{2\pi f_c x i}{c N_D} \right). \quad (12)$$

Eq. (12) specifies a relation between the TRRS decay and the distance moved by the wrist with a stretched arm. For $N_D = 50$, the RHS is plotted and is as shown in Fig. 5 indicating a monotonous decay. We exploit this monotonous decay to develop a likelihood-based approach for gesture classification in the next section.

IV. GWRITE ALGORITHM

This section designs a pipeline to extract unique features for each gesture shape and use them for classifying gestures. We consider gestures consisting of only straight-line segments in this work. It is sufficient to know the number of segments, the angle between two adjacent segments, and the locations of the intersection points between line segments, if any, to describe any shape composed of straight-line segments. An overview of

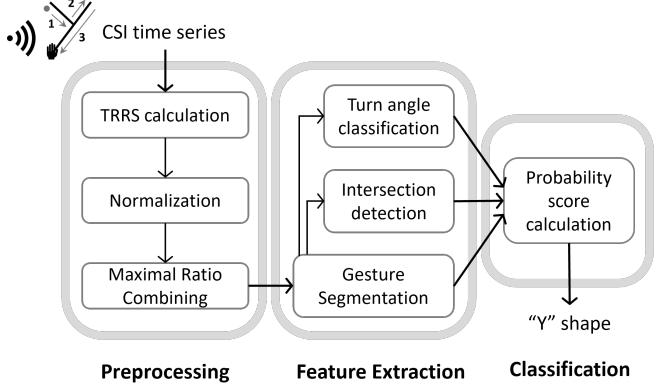


Fig. 6: Overview of *GWrite* algorithm.

GWrite module is shown in Fig. 6. Since we do not have the information about the absolute/relative direction of motion of the hand, we focus only on the relative shape of the trajectory. Thus, the features for the trajectories of letters 'N' and 'Z' will be the same with three line segments, acute angles between adjacent segments, and no intersection points. Similarly, the trajectory features for 'M' and 'W' will be the same. Sample trajectories from the upper case English alphabets with unique features are shown in Fig. 8. The *GWrite* prototype proposed in this work will demonstrate gesture classification on this set of 15 gesture shapes.

A. Preprocessing

The *GWrite* prototype is built on a WiFi chipset equipped with 2x2 MIMO creating four different spatial links in total. In the preprocessing step, we calculate the η between all pairs of CSIs and generate the TRRS matrix (η_z) for each link z . We then combine the matrices from all the links using MRC, and the resultant matrix will be used as an input for the feature extraction module.

- TRRS calculation:** For each link z , the $(i, j)^{\text{th}}$ entry of η_z is the TRRS between i^{th} and j^{th} CSI in the time series. For two CSIs H_i and H_j , $\eta_z(i, j)$ is defined in the frequency domain as [28],

$$\eta_z(i, j) = \frac{|\sum_{k \in \nu} H_i[k] H_j[k]^*|}{\sqrt{\sum_k |H_i[k]|^2 \sum_k |H_j[k]|^2}}, \quad (13)$$

where ν is the set of subcarriers and $(.)^*$ is the complex conjugate operation. One realization of η_z with $z = 1, 2, 3, 4$ is shown in Fig. 7a.

- Normalization:** Notice that T_S in (9), denotes the contribution of the static scatterers to the TRRS value. Different links could have different levels of T_S based on their multipath propagation. For each link, T_S is determined as the lowest value of TRRS among all the pairs of CSIs corresponding to a gesture. As T_S decreases, we observe a larger change in the TRRS for the same degree of hand movement. By subtracting the contribution from the static environment, we equalize the signal level on different

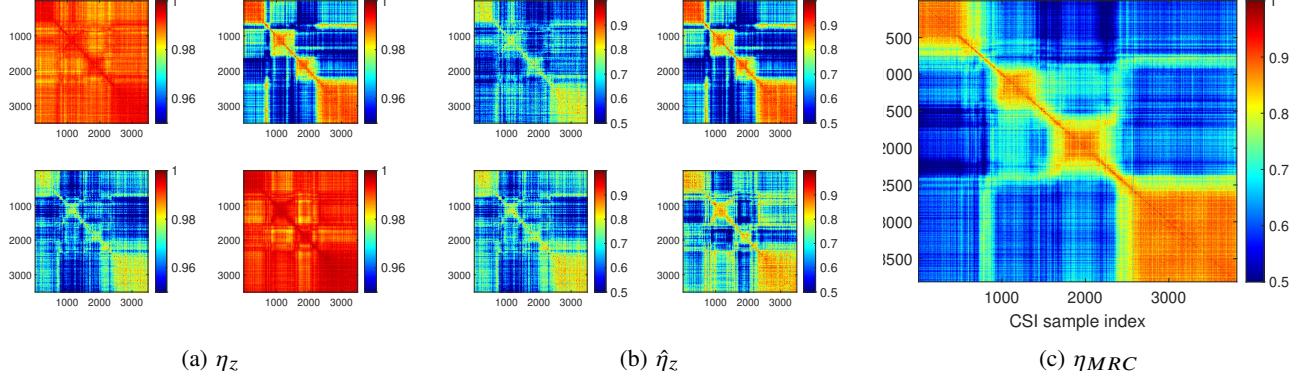


Fig. 7: Illustration of MRC of four TRRS links, $z = 1, 2, 3, 4$. (a) Raw TRRS matrices, (b) TRRS matrices after normalizing the static offset, and (c) TRRS matrix after MRC.

links. The normalized matrices will be denoted by $\hat{\eta}^z$ (Fig. 7b) and defined as,

$$\hat{\eta}_z = \frac{\eta_z - T_S}{1 - T_S}. \quad (14)$$

- **MRC:** It can be observed that different links present information of different significance due to the spatial diversity presented by the MIMO system. We combine the information from different links using the MRC technique by weighing them according to their signal and noise levels. When the speed of the hand is zero at the location of the turns, the η between CSIs should be equal to 1. This may not be the case in reality due to the environmental factors (noise due to ambient motion) and human factors (minor motions in other body parts/hand). Let this value of maximum η between CSIs, when the hand is at the turn location be denoted by η_z^{\max} , for each link z . The offset of the average η value at these zero speed instances from 1 indicates the noise level. The MRC weights are taken as inversely proportional to the square of the noise level. The combined η^{MRC} is then given by,

$$\eta^{\text{MRC}} = \frac{\sum_{z=1}^{N_{\text{TX}} * N_{\text{RX}}} \frac{\hat{\eta}(z)}{(1 - \eta_z^{\max})^2}}{\sum_{z=1}^{N_{\text{TX}} * N_{\text{RX}}} \frac{1}{(1 - \eta_z^{\max})^2}}, \quad (15)$$

where N_{TX} and N_{RX} denotes the number of Tx and Rx antenna respectively. η^{MRC} shows more significant variation in its values for a hand gesture and has a reduced noise level as shown in Fig. 7c. In the following feature extraction module, η^{MRC} will be analyzed to extract differentiating features for gesture shapes. In the following discussion, TRRS between two points in space should be understood as the TRRS between the CSIs with the hand at those spatial locations.

B. Feature Extraction

This module extracts the features characteristic of the gesture trajectory which will be later used for gesture classification.

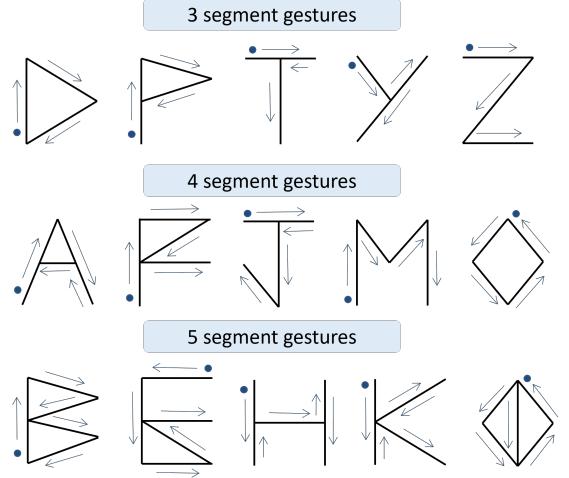


Fig. 8: Gesture trajectories of upper case English characters with different number of segments. The circular dot indicates the initial point for each shape.

1) *Gesture Segmentation:* Each straight line in a gesture shape is termed a segment. For example, different gesture shapes are shown in Fig. 8 which are grouped based on the number of segments (N_s). Gesture segmentation aims to determine the number of segments in a given gesture shape.

During a gesture, the speed is minimal when the hand changes the direction of motion, i.e., at the turns. Assuming the user performs the gesture with reduced/zero speed during turns, the number of turns can be inferred from the number of instances at which the speed is very low/zero. We leverage the motion statistics (MS) proposed in [29] to achieve this goal.

MS represent the extent of motion/disturbance in the wireless channel as perceived by the transceivers. As a result, a small dynamic object close to a transceiver and a larger dynamic object farther away from the transceiver could cause a similar amount of disturbance and result in the same MS . Similarly, a small object in rapid motion and a larger object with slower motion could have the same MS . MS are calculated from the correlation of the CSIs as detailed in [29]. The theoretical values lie between -1 and 1, with 1 indicating the

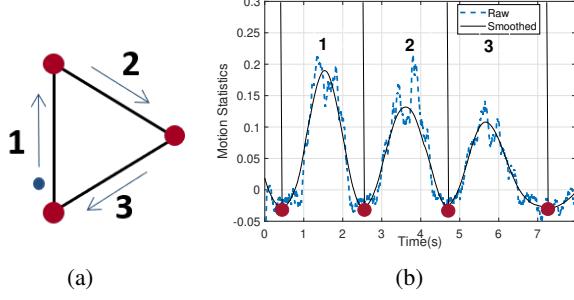


Fig. 9: Illustration of Gesture segmentation. (a) ‘D’ shaped gesture trajectory, and (b) the corresponding *MS* along with the different gesture segments.

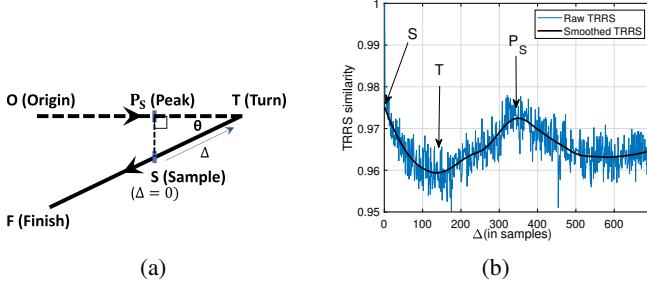


Fig. 10: (a) Geometry of a turn in a gesture, (b) \mathbf{S}^Δ plot showing the TRRS decay feature.

highest motion/disturbance level.

During a gesture, the person is at a fixed distance from the transceivers. Therefore, the *MS* can capture the relative rapidness of the hand motion. Consider a ‘D’ shaped gesture shown in Fig. 9a and the corresponding *MS* shown in Fig. 9b. Observe that the *MS* are lower at the location of the turns as marked in circles in Fig. 9. The number of straight-line segments in a gesture can be determined from the number of local minima/valleys in the *MS* minus 1.

2) *Turn Angle Classification*: A sudden change in the direction of movement of the hand is referred to as a “turn” event. For instance, Fig. 10a shows a gesture trajectory beginning at the Origin (‘O’), following a straight line till a “turn” occurs at the point ‘T’. The trajectory after the point ‘T’ is a straight line that ends at the point, ‘F’. The angle between the two straight line segments is referred to as the “turn angle”, θ . We classify θ into three groups/classes, namely, (i) $\theta \approx 0^\circ$, (ii) θ is acute i.e. $0^\circ < \theta < 90^\circ$, and (iii) $\theta \geq 90^\circ$. The rationale behind the choice of the angles is two-fold. Firstly, it enables the users to perform the turns efficiently without paying much attention to the exact angle drawn by the hand. Secondly, these groups of angles have unique differentiating features that enable classification based on the TRRS similarity, as we see below.

Consider any point ‘S’ on the second segment of the trajectory in Fig. 10a. Point ‘P_S’ is the closest point to ‘S’ on the first segment. If we compute the TRRS similarity between the CSI at the point ‘S’ and all the CSIs along the trajectory, moving backward, we observe a valley for the CSI at the point

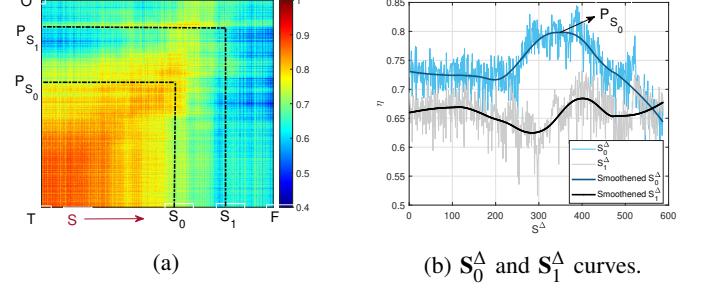


Fig. 11: Illustration of significance of peak with distance from the turn.(a) TRRS between two segments with an acute angle θ between them, and (b) \mathbf{S}^Δ curves for two points \mathbf{S}_0 and \mathbf{S}_1 on the second segment.

‘T’ and a peak for the CSI at the point ‘P_S’ as shown in Fig. 10b. Let us call this plot as \mathbf{S}^Δ for any point ‘S’ on the second segment. The direct correspondence between the relative distance and the TRRS is a result of the monotonous decay pattern derived in Section III. Leveraging the relative pairwise TRRS relationships between ‘S’, ‘P_S’ and ‘T’, we define two features, r and f , and use them to assign probability scores for different θ .

The parameter f :

The monotonic decrease of the TRRS similarity saturates beyond certain distance. As ‘S’ moves towards ‘F’ on the second segment (Fig. 10a), the distance between ‘S’ and the corresponding ‘P_S’ increases. As a result, the peak prominence in the \mathbf{S}^Δ plot continues to decrease and becomes negligible beyond a point. Fig. 11a shows the TRRS matrix between two segments with an acute angle θ between them. The \mathbf{S}^Δ plots for two points \mathbf{S}_0 and \mathbf{S}_1 on the second segment are shown in Fig. 11b. While the peak corresponding to \mathbf{S}_0 is significant, the peak corresponding to \mathbf{S}_1 is not obvious. Similarly, for a fixed ‘S’, as θ increases, the peak in \mathbf{S}^Δ plot becomes less significant and may disappear for larger θ . This observation motivated us to design the feature f for angle classification. It is defined as the fraction of the second segment for which a peak can be observed on the \mathbf{S}^Δ plot. If ‘ S_f ’ is the farthest point on the second segment from ‘T’, for which a prominent peak is observed on the \mathbf{S}^Δ plot, then f is defined by (distance between TS_f)/(distance between TF).

Given that each segment follows a straight-line path, the distance moved between any two instances/points on a segment can be determined as a fraction of the total length of the segment. Recall that *MS* were used in Section IV-B1 to determine the relative speed of the gesture at every instance. Assuming that the CSIs are equally spaced in time, we can use the *MS* as an approximate indicator of the relative distance moved by accumulating the *MS* between any two time instances. The approximation holds because a similar degree in the movement will cause similar *MS* within a single gesture realization. The relative distance between two points X and Y on a segment AB is obtained as a fraction of the length AB

TABLE I: Theoretical values of f and r for different θ .

Turn angle	r	f
$\theta \approx 0^\circ$	0	1
$\theta \approx 45^\circ$	1	1
$\theta \approx 90^\circ$	Undefined	0

by,

$$\frac{\sum_{i \in (X,Y)} MS(i)}{\sum_{i \in (A,B)} MS(i)}. \quad (16)$$

Then, f can be defined as:

$$f = \frac{\sum_{i \in (T,S_f)} MS(i)}{\sum_{i \in (T,F)} MS(i)}. \quad (17)$$

In this work, $f \leq 0.2$ is assigned to $\theta \geq 90^\circ$, and f should be equal to 1 for $\theta \approx 0^\circ$.

The parameter r :

For each point ‘S’ on the second segment, the corresponding points ‘T’ and ‘P_S’ can be obtained from the S^A curve. For a given turn angle θ , the parameter r is defined as the ratio of areas between the curves $T_{max} - TRRS(S, P_S)$ and $T_{max} - TRRS(S, T)$, where T_{max} is the highest TRRS between points on the two segments. Ideally $T_{max} \approx 1$ as the TRRS between points close to the point ‘T’ lying on different segments is close to 1. If $\theta = 45^\circ$, $SP = PT$ and assuming a uniform environment, $TRRS(S, P_S) \approx TRRS(P_S, T)$ resulting in $r \approx 1$. For $\theta = 0^\circ$, $TRRS(S, P_S) \approx T_{max}$ resulting in $r \approx 0$.

The ideal values of the parameters f and r are shown in the Table I. However, the observed values differ from their ideal values due to the following:

- The monotonous decay of the TRRS is not evident over a large distance. Therefore, for larger acute angles, the peaks may not be detected for the entire length of the second segment and $f < 1$.
- $TRRS(S, P_S)$ is not always equal to 1 for $\theta \approx 0^\circ$. This is because the human body and the hand are non-rigid objects and the CSI may not match exactly even though the person intends to trace the same path. Therefore, $r > 0$ for $\theta \approx 0^\circ$.
- Since there is no reference when drawing gestures in the air, it may be difficult for the user to make a precise 45° turn. Therefore, the value of r has a wide range of values for an acute angle turn.

We use a data-driven approach to obtain the distribution of the parameter r and design a probability score for the angle classification. For this, we gathered CSI data corresponding to 400 turns each, for the three categories of angles and computed r . The empirical distribution of r is shown in Fig. 12. The theoretical values of r for $\theta = 0^\circ$ and $\theta = 45^\circ$ are 0 and 1 respectively. However, we do not restrict the user to perform a strict 45° turn and allow any acute angle, resulting in a wider spread of r for the later class. The classification threshold is determined as the one which achieves the highest classification accuracy (0.35). The probability score curve for $\theta \approx 0^\circ$ is defined as the half-normal density centered at $r = 0$. The curves are defined such that they sum to 1 and intersect at

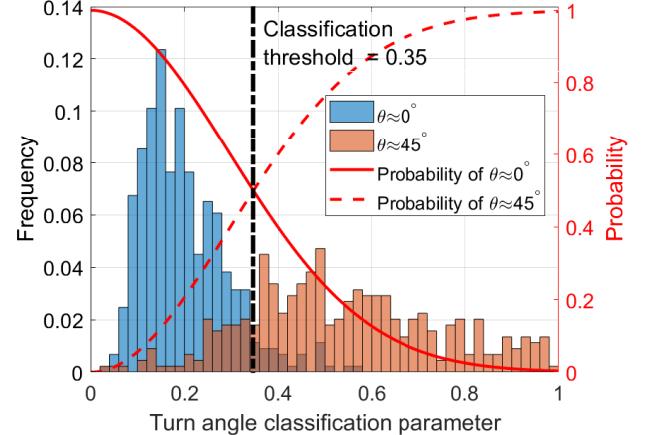


Fig. 12: Distribution of the parameter r for $\theta \approx 0^\circ$ and $\theta \approx 45^\circ$.

the threshold value (0.35) with a probability score of 0.5 as shown in Fig. 12. The relative values of the probability scores will enable us to calculate a probability score for each gesture shape for classification.

3) *Intersection Point Detection*: Identifying and locating intersections between two non-adjacent line segments can be a useful feature to differentiate gesture shapes. For instance, the third segment intersects the first segment at the mid-point and at the start point in ‘P’ and ‘D’ shapes respectively. If an intersection/crossing occurs in a gesture trajectory, it implies that the spatial orientation/location of the arm is almost the same at those two time instances, resulting in a relatively high TRRS value between the two CSIs (ideally equal to 1). However, occurrence and the location of the intersection point are not the only deciding factors for gesture classification. Defining a probability score could be more informative for the overall classification rather than a binary decision on the intersection. There exist two main challenges ahead.

- It is evident that if an intersection occurs, it will be the location(instance corresponding to the maximum TRRS among all the pair-wise TRRS similarities between the CSIs from the two segments. However, we do not know the absolute distance moved between any two time instances, which makes the estimation of intersection location non-trivial.
- Although the value of TRRS similarity at an intersection point is ideally equal to 1, the environmental and human factors decrease the value to less than 1. The influence of these factors being unpredictable and difficult to model makes the probability score assignment more involved.

Observe that we only need a relative location of the intersection point on each of the two segments under consideration. As earlier, we leverage the accumulated MS as a measure of distance between two points. Consider a P-shaped gesture shown in Fig. 13. Let us denote the point of intersection on a segment i by the fraction of its distance to the total length from the start point of the segment using ϕ_{pi} . The intersection point between Segment 1 and Segment 3 is $(\phi_1, \phi_3) = (0.5, 1)$ and is marked by the darker circle. Since the user does not have any reference of the spatial trajectory drawn till then, it is difficult

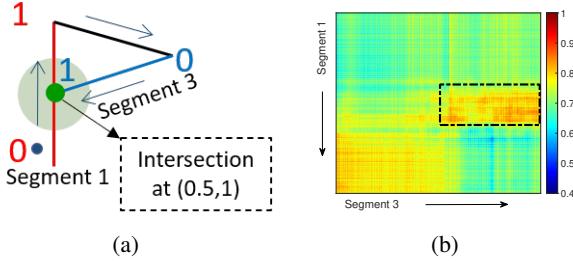


Fig. 13: (a) Illustration of the intersection point detection, (b) TRRS between CSIs of segment 1 and segment 3.

to cross segments at the desired location. We, therefore, allow a margin of 0.25 as indicated by the extended circular region in Fig. 13. If the intersection point lies within the margin of 0.25 of the desired location on both the segments, we estimate a probability score for an intersection point.

We use a similar approach to that in Section IV-B2 for the probability score assignment. We design a heuristic $T_N = \frac{T_{\max} - T_{\min}}{1 - T_{\min}}$, where T_{\max} and T_{\min} are the maximum and minimum values of $TRRS(i, j)$ with i and j lying on the two segments that could possibly have an intersection. Fig. 14 shows a histogram of T_N values for 600 pairs of segments each, with and without an intersection. The value of T_N for an intersection point is ideally equal to 1. From the histogram, the classification threshold of 0.71 achieves the highest classification accuracy for intersection vs no-intersection between two segments. As mentioned before, instead of a binary classification, a probability score will retain more information and will be used in the final step of gesture classification. The probability curve for the ‘intersection point’ class is defined as the half-normal density centered at 1 such that the probability score value is 0.5 at threshold (0.71) as shown in Fig. 14. The probability curve for the ‘no intersection’ class is taken as one minus the probability of intersection. In the extreme case of T_N being very low, i.e., if the two segments are far apart, then the probability of an intersection point is equal to 0.

C. Gesture Classification

Gestures can be classified using the probability scores of the features calculated in the feature extraction module. First, the gesture group is decided based on the number of segments. The probability score for each character (Ω) within the group is then calculated as follows:

$$P(\Omega) = \frac{\alpha}{N_s - 1} \sum_{i=1}^{N_s-1} P(\theta_i^\Omega) + \frac{(1-\alpha)}{N_s-1 C_2} \sum_{i=1}^{N_s-1} \sum_{j=i+2}^{N_s} P(\{\phi_i, \phi_j\}^\Omega), \quad (18)$$

where N_s is the total number of segments in the gesture trajectory, $N_s-1 C_2$ is the combination notation indicating the total number of pairs of non-successive segments that can potentially intersect. Notice that j is indexed from $i + 2$, because adjacent segments cannot intersect other than at the turn locations. The actual values of the features for the trajectory shape Ω are given by θ_i^Ω and $\{\phi_i, \phi_j\}^\Omega$. For example, $\{\theta_1^\Omega, \theta_2^\Omega, \{\phi_1, \phi_3\}^\Omega\} = \{45^\circ, 45^\circ, \{0.5, 1\}\}$. The parameter α is

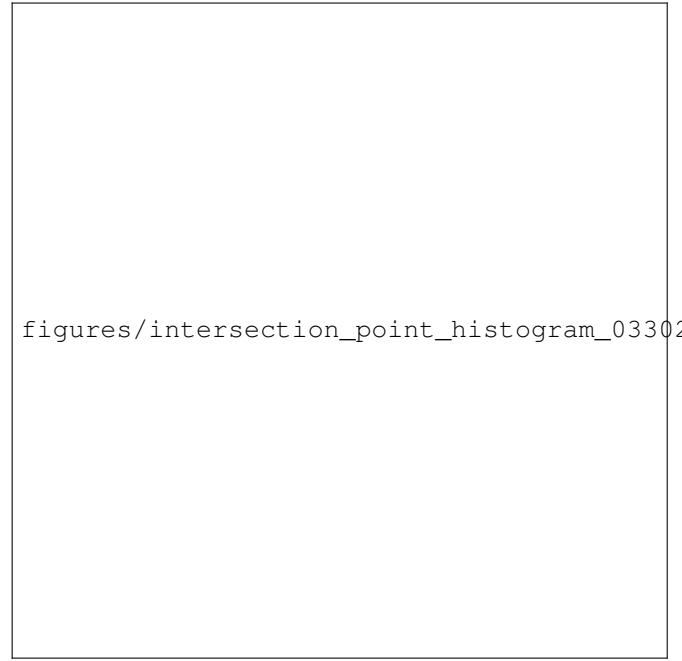


Fig. 14: Histogram of T_N values with and without an intersection and the corresponding probability score assignment.

TABLE II: Probability scores for characters with 3 segments.

Ω	Probability score
D	0.42
P	0.82
T	0.17
Y	0.32
Z	0.42

a hyperparameter that can be tuned to weigh the contribution of the angle classification and intersection point detection modules. Table II shows an example of the calculated probability scores for a P-shaped gesture, where $\alpha = 0.5$ and $P(\{\phi_1, \phi_3\}) = 0.8$. The probability score for $\Omega = \text{P}$ is the highest (0.82) amongst others and the gesture shape can therefore be classified as a P-shape.

V. PERFORMANCE EVALUATION

To evaluate the performance, we built a prototype of *GWrite* on off-the-shelf commercial WiFi chipsets with a bandwidth of 80 MHz, 2x2 MIMO using a sampling rate of 350 Hz. A typical through-the-wall setup is as shown in Fig. 1. The transmitter (Tx) and the receiver (Rx) are placed in different rooms. Gestures are performed in a different room with at least one wall between each transceiver and the gesture location.

A. Performance of Feature Extraction

1) *Gesture Segmentation Accuracy*: We have collected 100 samples of each of the gestures with 3, 4 and 5 segments for evaluation. The number of segments was correctly identified 100% of the time. Since the gesture segmentation module can accurately estimate the number of segments in a gesture trajectory, in the gesture classification accuracy evaluation,



Fig. 15: Confusion matrices for (a) angle classification and (b) intersection point detection.

we evaluate classifying gestures with the same number of segments.

2) *Turn Angle Classification Accuracy*: The angle classification module discussed in Section IV-B2 is evaluated using about 400 gesture realizations each with $\theta \approx 0^\circ$ and $\theta \approx 45^\circ$. The confusion matrix of the classification is shown in Fig. 15a. The average classification accuracy is 90.4%.

3) *Intersection Point Detection Accuracy*: The intersection point detection module discussed in Section IV-B3 is evaluated with about 600 pairs of segments each with and without an intersection. The confusion matrix of the classification is as shown in Fig. 15b with an overall classification accuracy of 81.1%.

B. Gesture Classification Accuracy

For evaluating the overall performance of the gesture classification algorithm, we group sets of uppercase English characters with the same number of segments and perform gesture classification within each group. This is because the gesture segmentation is highly accurate and we found zero misclassification based on the number of segments. We divide the characters with unique gesture shapes into three groups as shown in Fig. 8 into,

- 3 segments: D, P, T, Y and Z shapes.
- 4 segments: A, F, J, M and O shapes.
- 5 segments: B, E, H, K and Q shapes.

The confusion matrix for each group is as shown in Fig. 16. The average classification accuracies for 3-, 4- and 5-segment groups are 90.8%, 88.4% , and 96.8% respectively. The higher classification accuracy for 5 segment gestures is because of more differentiating features which decreases the probability of misclassification. The average classification accuracy for all the 15 gestures is 92%. For applications requiring a smaller number of gestures, 5-segment gestures can be used whose classification accuracy is about 97%.

The data was collected over five months during off-work hours. Around 25 people work in the office space causing many changes to the indoor environment over a long period. Also, the Tx, Rx, and gesture locations are neither fixed nor marked. Instead, the devices are casually placed on a table around the locations marked on the floor map, unlike most existing works that cannot work even with a slight change in the Tx/Rx locations. We thus demonstrate that *GWrite* is robust to changes in the indoor environment. In VI-A, we also discuss

that the distribution of features for different classes follows the same trend in an entirely different experimental setup.

VI. DISCUSSIONS

A. Placement and Coverage

The transceiver placement is flexible with *GWrite* as the features for classification are characteristic of the trajectory shape rather than the geometry of the placement. While the exact placement is irrelevant, there is a specific operation region in which *GWrite* can work best. If the transceivers are placed too far away from each other, then the channel perturbation caused due to the movement of the hand may not be perceived. On the other hand, if the devices are placed very close to each other, then the uniform scattering assumption may not hold, and the performance might degrade. We show a few experimental setups in Fig. 17 suitable for *GWrite*.

The term T_S in (12) denotes the extent of the contribution from the static environment to the TRRS and is dependent on the relative location of the transceivers, gesture and the environment. We compensate for T_S while combining the TRRS from different links in the preprocessing stage. As a result, the features for angle classification and intersection detection are independent of the location/distance between the transceivers within the region of operation. Fig. 18 shows the histograms of T_N for two locations. It can be observed that the classification thresholds are similar in both cases based on which we design the probability score curves. Location independence is one of the crucial advantages provided by *GWrite*.

B. Weightage of Features (α)

The parameter α in (18) can be adjusted to give different weightage to the angle classification and the intersection point detection modules. The optimal value depends on the set of gestures to be classified. For example, in the extreme case of using a gesture set consisting only of shapes with acute angles, α can be set to 0, in which case the classification is purely based on the intersection detection module. The average classification accuracy of 3-segment gestures for different α is shown in Fig. 19.

C. Generalized Gestures

We evaluated the performance of *GWrite* on a set of 15 upper case English characters. However, this set can be expanded to include generalized shapes with straight lines, such as in Fig. 20. *GWrite* can classify gestures composed of straight-line segments that differ in terms of the turn angles and the intersection points. The more the number of such differentiating features, the higher the classification accuracy.

D. User diversity

Relying on the relative shape of the gesture, we do not expect noticeable user diversity with *GWrite*. To confirm this assertion, we have conducted experiments with six different users whose information is given in Table III. Each user is asked to repeat the 15 gestures 30 times each, and the gathered

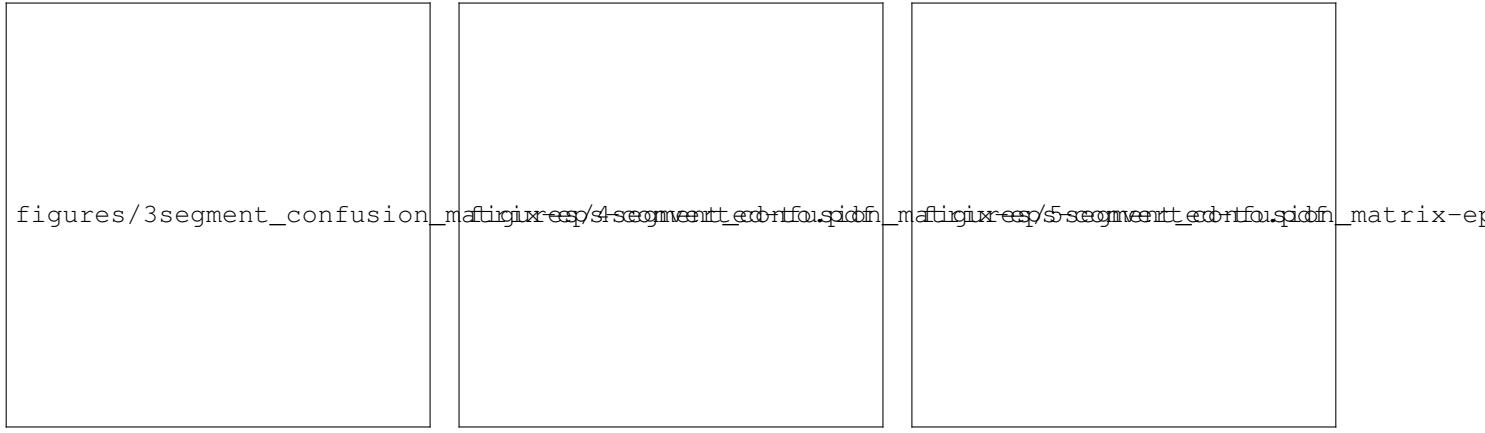


Fig. 16: Confusion matrices for classification of gestures with different number of segments.

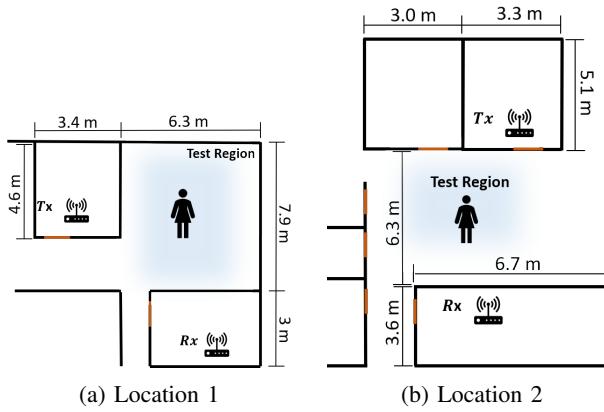


Fig. 17: Examples of device placement.

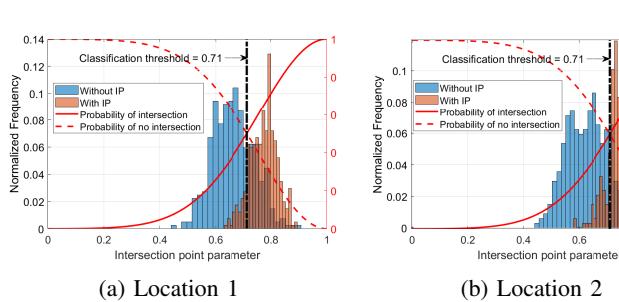


Fig. 18: Histograms of T_N for two different device placements.

CSI time series is used for gesture classification. Fig. 21 shows a similar distribution of the parameters r and T_N for user 2. Applying the thresholds determined by the first user, we achieved a classification accuracy of 89.3% on the set of 15 gestures. The angle and intersection point classification accuracy is 85.6% and 80.4%, respectively. We believe the normalization step in the TRRS calculation also helps to achieve user independence apart from location independence. The average classification accuracies ranged between 83% and 92% as shown in Fig. 22 for all the users. A major part of the classification accuracy difference can be attributed to how different users draw gestures, especially the intersections and

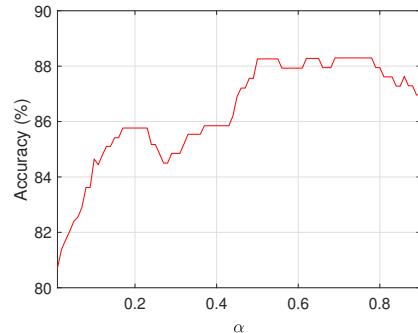


Fig. 19: Classification accuracy with α .

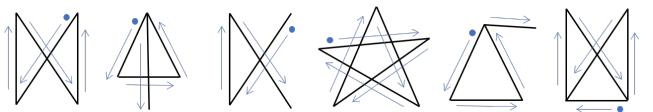


Fig. 20: More shapes that can be classified with *GWrite*.

the turn angles. From this data, we do not observe much user dependency. However, an extensive large-scale study is needed to fully understand the performance with more users and over time with the same user when he/she gets comfortable/trained to draw the gesture shapes.

TABLE III: User Information

User	Height (cm)	Weight (kg)	Age	Gender	Arm length (cm)
1	164	52	29	F	63
2	165	50	26	F	53
3	172	72	27	M	56
4	169	56.7	25	F	55
5	158	50	28	F	52
6	173	90	29	M	60

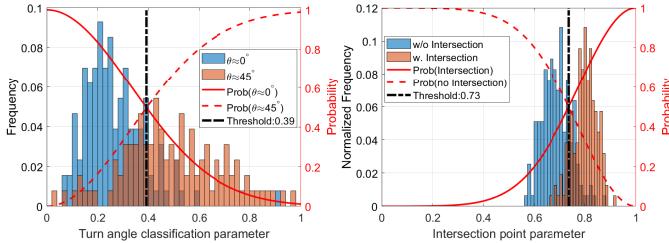
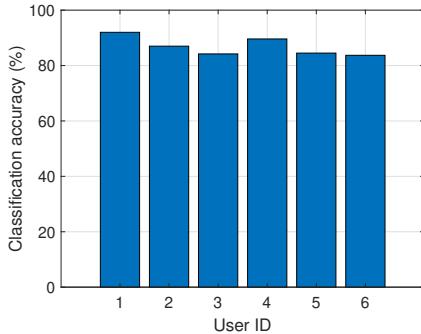
Fig. 21: Distribution of (a) r , and (b) T_N for a new user.

Fig. 22: Classification accuracy for different users

E. Relaxed gestures

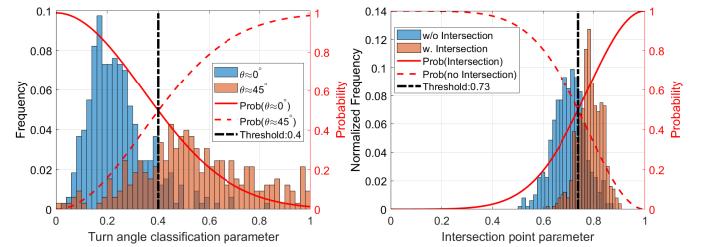
In Section III, the TRRS relation between \mathbf{h}_0 and \mathbf{h}_x in (12) was derived under the assumption that the hand is completely stretched out while performing the gesture as shown in Fig. 23a. However, it might be a bit different than the natural way of writing in the air with a relaxed arm, as shown in Fig. 23b. To gauge the performance under such a relaxed scenario, we evaluated the performance of *GWrite* over the same set of 15 gestures without any constraint on the arm. The distribution of the parameters r and T_N is shown in Fig. 24. The angle and intersection point classification accuracies are 86.52% and 79.09% respectively while the overall gesture classification accuracy is 74.16%. Although the accuracy is lower than that of the stretched hand method, it is high and promising for a sizable set of 15 gestures.

F. Computational complexity

Gesture recognition systems are practical when implemented in real-time, enabling a smooth user experience. The highest complex steps in *GWrite* are the TRRS matrix computation, smoothing operations on S^Δ plots, and peak detection. These operations are $O(n^2)$ complex where n is the length of the CSI time series. For instance, if the gesture duration is 3 seconds and the CSI sampling rate is 350 Hz, then the TRRS matrix requires calculating the TRRS metric 1050 X 1050 times. It takes less than 5 seconds to determine the gesture shape from the CSI time series on a Windows computer (Intel Core i7-7700K @4.2GHz, RAM 32 GB, 64-bit). With improved algorithm optimizations and the availability of GPUs, we believe *GWrite* can be implemented in real-time.



Fig. 23: Gesture writing with (a) Stretched arm, and (b) Relaxed arm.

Fig. 24: Distribution of r and T_N for gestures with a relaxed arm.

G. Background motion interference

The proposed *GWrite* system is robust to the presence of interfering users' motion as long as the motion is relatively farther away from the user performing the gesture. We have conducted experiments with one user performing the gesture and another person walking in the background to understand the level of interfering motion with distance from the user performing the gesture. Fig. 25 shows the experimental setup where d is the distance between the gesture location and the walking person. The extent of interference created by the walking human can be visualized in terms of the motion statistics level as shown in Fig. 26 with different distances from the user. Motion closer than 6m is high enough to disrupt the gesture segmentation, as no clear valleys were found in the motion statistics curve (Sec. IV-B1). Also, performance degradation can be observed because the matching points and turn angle features are distorted due to the background motion for samples with the correct number of segments. For example, in an experiment involving a second user walking in the background, we achieved an accuracy of 86% and 67% where d is 6m and 5m, respectively. In general, *GWrite* can work without much performance degradation, even when the interfering human is sitting/working at a desk or has high motion farther from the Tx-Rx link and the gesture location. Learning techniques might help to improve gesture segmentation by choosing adaptive parameters based on the background motion level. We leave this study for future work.

H. Comparison with state-of-the-art

CSI has been used in previous works to realize gesture recognition. In table. IV, we compared different CSI-based

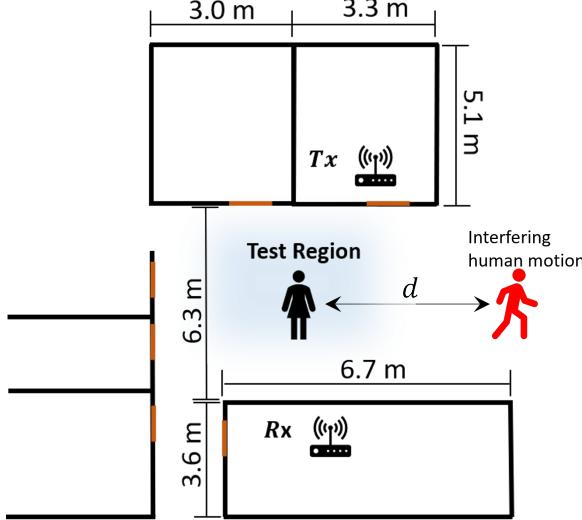


Fig. 25: Experiment setup for background motion interference test.

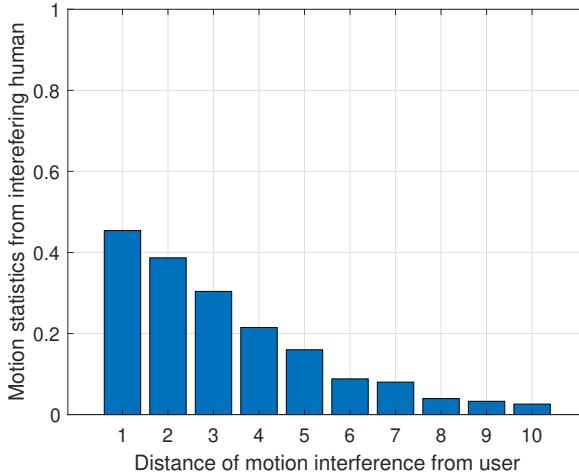


Fig. 26: Level of motion statistics induced when the interfering person is moving away from the user.

gesture recognition works in terms of the features/method used, experimental setup, gesture set, and performance. For example, the entry $iTx-jRx$ in the third column indicates that i Tx and j Rx are required in that work. Also, the “Device-User” indicates the distance between the user and one of the transceivers for the system to work. The fifth column compares in-domain and cross-domain accuracies, which measure the extent of location independence achieved by the respective features. Finally, the last column indicates if the system needs to be re-trained for a new environment/setup/gesture or user orientation. From the table, only WiGest [30], EI [31], and *GWrite* have achieved location independence to different extents, while other works require at least a few training samples for a new location. Below are a couple more observations from the table.

- Achieving location-independent gesture recognition is challenging, and previous attempts have reported a drasti-

cally reduced cross-domain accuracy compared to the in-domain accuracy. For example, WiMU [32] reported an accuracy drop from 96% to 85% just for a change in the user orientation, even for the same Tx and Rx location. *GWrite* is the only work to report accuracy above 90% even for cross-domain, thanks to the location-independent features.

- Amongst the three location-independent works reported in the table, *GWrite* has the broadest coverage, highest Device-User distance, most extensive gesture set that is further expandable, and a good classification accuracy.
- The deep learning-based environment dependency removal can tolerate only a certain amount of environment shift, such as the one in the in-car driver authentication [33] and other works that claim environment independence [31]. The relative device locations of the target performing the activity/gesture are similar, with a short device-to-user distance enabling a strong gesture signal. On the other hand, *GWrite* demonstrates a much higher degree of location independence even with a larger device-to-user distance and tolerating indoor environment changes over a long period of time, as discussed in V-B.

VII. FUTURE WORK

As discussed in the previous section, achieving robust through-the-wall location independent gesture recognition is challenging and we have shown that *GWrite* is a step ahead in achieving it along with relaxed device placement requirements. When more data is accumulated over time, following are a few things we would like to explore in our future work.

- In this work, we have kept the feature extraction and gesture classification simple and intuitive to make the system more robust. The next step would be to try machine learning classifiers or CNN for pattern recognition using the current performance as a baseline.
- More experiments are needed to fully understand the performance/limitations of *GWrite* under different setups and users and over time.

VIII. CONCLUSION

We have proposed *GWrite*, a device-free gesture writing recognition system using commercial WiFi devices. The proposed system can work in through-the-wall scenarios and over a broad set of gestures. Using a gesture model and embracing the intense multipath in a typical indoor environment, we have proved that the TRRS between CSIs decreases monotonically with the distance moved by the hand. This observation allowed us to extract features describing geometric relationships in gesture shapes consisting of straight-line segments. We have evaluated *GWrite* on a set of 15 uppercase English characters and achieved a recognition accuracy of 92%. As more bandwidth becomes available with WiFi 6/7, the real-world gesture writing comply more closely with the theoretical model of *GWrite*, and *GWrite* can achieve higher recognition accuracy, playing an integral role in the future of smart indoor environments and HCI applications.

TABLE IV: Comparison of state-of-the-art WiFi-based device-free CSI-based gesture/activity recognition systems

Reference	Method	Placement	No. and type of gestures	Claimed Accuracy	Training-free
WiG [34]	SVM	1Tx-1Rx: 1-3m, Device-User: < 1m	4: right, left, push, pull	92% LOS, 88% NLOS	✗
WiGeR [20]	Wavelet transform, DTW	1Tx-1Rx: 3.5m, Device-User: 2m	7: left, right, flick, grab, up, down, point	97% LOS, 84% NLOS	✗
CARM [35]	DFS profiles, HMM model	1Tx-1Rx: 3.5 m, Device-User: 1.7m	8: running, walking, boxing, brushing, pushing, falling etc.	96% in-domain, 85% cross-domain	✗
WiMU [32]	Virtual sample generation, STFT	1Tx-1Rx: 1.5m, Device-User: 3m	6: open/close door, sit/stand, circular arm, kicking, etc.	95% in LOS, 89.5% in NLOS	✗
WiHF [36]	DFS spectrogram, DNN for gesture and user ID	1Tx-3Rx: 2m, Device-User: 1m	10: numbers 0-9	94% in-domain, 90% cross-domain	✗
WiDar 3.0 [37]	BVP profile, DFS spectrum, CNN, GRU	1Tx-3Rx: 2m, Device-User: 1m	10: numbers 0-9, push, sweep, clap, slide, circle,zizag	92% in-domain, 83% cross-domain	✗
Cross Sense [38]	ANN,mixture-of-experts model, ACF, spectrogram	1Tx-1Rx: 1m, Device-User: 0.5 m	40: flick, punch, bowling, swipe, sit, brush, cooking,etc.	95% for 15 gestures	✗
WiGest [30]	Wavelet transform, gesture families	Max 1Tx-3Rx: 8m, Device-User: 1.2m	7: combination of far/near, slow/fast, pausing	LOS and NLOS; 88% - single AP, 96% - 3 APs	✓
EI [31]	Lagged correlations between segments, adversarial network	1Tx-2Rx: 2m, User-Device: 0.5m	6: wiping board, walking, moving, rotating chair etc.	75% cross-domain	✓
GWrite	TRRS, angle and intersections b/w segments	1Tx-1Rx: 13m, Device-User: 6m	15: Upper case english alphabet with straight lines	Through-wall, in-domain and cross-domain: 92%	✓

Note: Values rounded to nearest integer.

REFERENCES

- [1] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 235–246, 2014.
- [2] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 15–28.
- [3] B. Wang, Q. Xu, C. Chen, F. Zhang, and K. J. R. Liu, "The promise of radio analytics: A future paradigm of wireless positioning, tracking, and sensing," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 59–80, 2018.
- [4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [5] S. D. Regani, Y. Hu, B. Wang, and K. R. Liu, "Wifi-based robust indoor localization for daily activity monitoring," in *Proceedings of the 1st ACM Workshop on Mobile and Wireless Sensing for Smart Healthcare*, 2022, pp. 1–6.
- [6] X. Zeng, B. Wang, C. Wu, S. D. Regani, and K. R. Liu, "Wicpd: Wireless child presence detection system for smart cars," *IEEE Internet of Things Journal*, 2022.
- [7] Y. Hu, M. Z. Ozturk, F. Zhang, B. Wang, and K. R. Liu, "Robust device-free proximity detection using wifi," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7918–7922.
- [8] F. Zhang, C. Chen, B. Wang, and K. J. R. Liu, "WiSpeed: A statistical electromagnetic approach for device-free indoor speed estimation," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2163–2177, 2018.
- [9] D. Wu, R. Gao, Y. Zeng, J. Liu, L. Wang, T. Gu, and D. Zhang, "FingerDraw: Sub-wavelength Level Finger Motion Tracking with WiFi Signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–27, 2020.
- [10] M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-free passive localization for wireless environments," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 222–229.
- [11] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.
- [12] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang, "Radio-frequency tomography for passive indoor multitar get tracking," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2322–2333, 2012.
- [13] H. Abdelnasser, K. A. Harras, and M. Youssef, "Ubibreath: A ubiquitous non-invasive wifi-based breathing estimator," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2015, pp. 277–286.
- [14] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.
- [15] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.
- [16] S. Tan and J. Yang, "WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 201–210.
- [17] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, "WiFinger: Talk to your smart devices with finger-grained gesture," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 250–261.
- [18] W. He, K. Wu, Y. Zou, and Z. Ming, "WiG: WiFi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–7.
- [19] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 90–102.
- [20] M. A. A. Al-qaness and F. Li, "Wiger: Wifi-based gesture recognition system," *ISPRS International Journal of Geo-Information*, vol. 5, no. 6, p. 92, 2016.
- [21] H. Abdelnasser, K. Harras, and M. Youssef, "A ubiquitous WiFi-based fine-grained gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2474–2487, 2018.
- [22] S. D. Regani, B. Wang, M. Wu, and K. J. R. Liu, "Time Reversal Based Robust Gesture Recognition Using WiFi," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8309–8313.
- [23] Z. Tian, J. Wang, X. Yang, and M. Zhou, "WiCatch: A Wi-Fi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16911–16923, 2018.
- [24] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 77–89.
- [25] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 27–38.
- [26] F. Zhang, C. Chen, B. Wang, H.-Q. Lai, Y. Han, and K. J. R. Liu, "WiBall: A time-reversal focusing ball method for decimeter-accuracy indoor tracking," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4031–4041, 2018.

- [27] S. Thoen, L. Van der Perre, and M. Engels, "Modeling the channel time-variance for fixed wireless communications," *IEEE Communications letters*, vol. 6, no. 8, pp. 331–333, 2002.
- [28] C. Chen, Y. Chen, Y. Han, H.-Q. Lai, and K. J. R. Liu, "Achieving centimeter-accuracy indoor localization on wifi platforms: A frequency hopping approach," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 111–121, 2016.
- [29] F. Zhang, C. Wu, B. Wang, H.-Q. Lai, Y. Han, and K. J. R. Liu, "WiDetect: Robust Motion Detection with a Statistical Electromagnetic Model," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–24, 2019.
- [30] H. Abdelnasser, K. Harras, and M. Youssef, "A ubiquitous wifi-based fine-grained gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2474–2487, 2018.
- [31] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsoukos *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th annual international conference on mobile computing and networking*, 2018, pp. 289–304.
- [32] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 401–413.
- [33] S. D. Regani, Q. Xu, B. Wang, M. Wu, and K. R. Liu, "Driver authentication for smart car using wireless sensing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2235–2246, 2019.
- [34] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–7.
- [35] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.
- [36] C. L. Li, M. Liu, and Z. Cao, "Wihf: Gesture and user recognition with wifi," *IEEE Transactions on Mobile Computing*, 2020.
- [37] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 313–325.
- [38] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "Crosssense: Towards cross-site and large-scale wifi sensing," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 305–320.