WiCGesture: Meta-Motion-Based Continuous Gesture Recognition With Wi-Fi

Ruiyang Gao, Wenwei Li, Jinyi Liu, Shuyu Dai, Mi Zhang, *Senior Member, IEEE*, Leye Wang, *Member, IEEE*, and Daqing Zhang, *Fellow, IEEE*

***Abstract*—Recent advancements in Wi-Fi-based sensing technologies have enabled effective hand gesture recognition. However, most studies focus on single gesture recognition and fail to recognize naturally performed continuous gestures without pauses in transitions. The main challenges include diverse and uncertain transitions in continuous gesture recognition, making it difficult to segment and identify gestures from a stream of continuous hand movements. In this article, we introduce a new method to recognize continuously performed gestures from a set of predefined gestures (e.g., digits) without requiring a pause in transitions. Instead of segmenting gestures at the gesture-transition level, we segment the stream into basic fractions that depict exclusive moving patterns of gestures. We propose a novel feature called *meta motion*, which geometrically characterizes different basic hand movements. Leveraging this feature, we use a back-tracking searching-based algorithm to identify gestures from the sequence of meta motions. Based on this approach, we develop a prototype system, *WiCGesture*, on commodity Wi-Fi devices. *WiCGesture* is the first system engaging in continuous gesture recognition using Wi-Fi signals. Evaluation results show that *WiCGesture* effectively recognizes continuous gestures from two gesture sets, significantly outperforming state-of-the-art methods.**

***Index Terms*—CSI, gesture recognition, Wi-Fi Sensing.**

I. Introduction

THE PAST decade witnessed the rapid development of Wi-Fi-based sensing technologies for human activities [1], [2], [3]. Given the superiority of Wi-Fi signals with its ubiquitousness and low cost, the recently merged Wi-Fi-based gesture recognition technologies [4], [5], [6], [7], [8], [9], [10] support the noncontact and natural method of obtaining the user's intentions through gestures in human–computer interaction (HCI).

In actual applications of gesture-based HCI, users can continuously perform gestures to coherently convey their intentions. For example, for handwriting applications, it is natural for users to write multiple letters without any pause. While recent Wi-Fi-based in-air gesture recognition methods can effectively identify various single-input gestures, they do not engage with recognizing gestures from the continuous stream of hand movements. Existing methods extract unique patterns [5], [9], [11], [12], [13], [14] to map different gestures or generate explicable features [7], [10], [15], [16], [17] by modeling hand movements for gesture recognition. However, input data fed into these classifiers is confined to an isolated single gesture. Most of them [5], [9], [11], [12], [13], [14], [15], [16], [17] are not capable of recognizing continuous gestures in a natural scene without any automated data segmentation scheme.

Some methods [7], [8], [10] segment gestures in real-time systems by determining the hand's status, specifically whether it is moving or static. They consider an action fragment lasting for a period as an input gesture, while stationary segments represent transitions between adjacent gestures. However, in certain HCI-based applications like handwriting, the hand may be static or continuously moving during transitions if the user naturally performs gestures.

When the hand moves between two consecutive gestures, there is no unique instantaneous feature to mark gesture boundaries and transitions. The hand's moving pattern during the transition is also unpredictable. As shown in Fig. 1, transitions between adjacent gestures (e.g., "2" and "3") can have different displacements and directions, leading to diverse signal changes. Methods that segment gestures by determining the moving-static state cannot guarantee capturing an entire gesture from continuous motion streams, thus failing to naturally recognize continuous gestures.

Furthermore, when a gesture is performed at different times or by different users, hand speed can vary, resulting in diverse durations. These variations pose significant robustness challenges. Pattern-based methods may need more new samples and extensive labor to segment data in the continuous gesture scenario. DSW [18] proposed a dynamic speed adaptive scheme to recognize gestures with speed variations; however, it requires a specific trajectory with identical start and end postures, which is not feasible for continuous gesture recognition.

In this article, we propose a novel gesture recognition scheme that identifies naturally performed continuous gestures, referring to gestures in a scenario where users do not pause during transitions. We propose that exact segmentation of

Manuscript received 25 May 2023; revised 24 October 2023; accepted 30 November 2023. Date of publication 18 December 2023; date of current version 25 April 2024. This work was supported in part by NSFC A3 under Project 62061146001, and in part by PKU-NTU Collaboration Project. *(Corresponding author: Daqing Zhang.)*

Ruiyang Gao, Wenwei Li, Jinyi Liu, Shuyu Dai, and Leye Wang are with the School of Computer Science, Peking University, Beijing 100871, China (e-mail: gry@pku.edu.cn).

Mi Zhang is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: mizhang.1@osu.edu).

Daqing Zhang is with the Department of Computer Science and Technology, School of Computer Science, Peking University, Beijing 100871, China, and also with the Telecom Sud-Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: dqzhang@sei.pku.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3343875

2327-4662 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON BETWEEN WICGESTURE AND STATE-OF-THE-ARTS

	Methodology	Key Technique	Gesture Recognition Mode
WiFinger [9]	Learning-based	CSI Variations (Time domain)	Single Input Gesture
WiDar3.0 [16]	Physical Model-based	BVP (Velocity-related)	Single Input Gesture
WiGesture [7]	Physical Model-based	MNP (Direction-related)	Require Pause between Adjacent Gestures
HandGest [10]	Physical Model-based	DPV and MRV (Direction-related)	Require Pause between Adjacent Gestures
DPSense [8]	Physical Model-based	EDP+MNP (Direction-related)	Require Pause between Adjacent Gestures
WiCGesture (ours)	Physical Model-based	Meta Motion (Displacement and Direction related)	Continuous Input Gestures

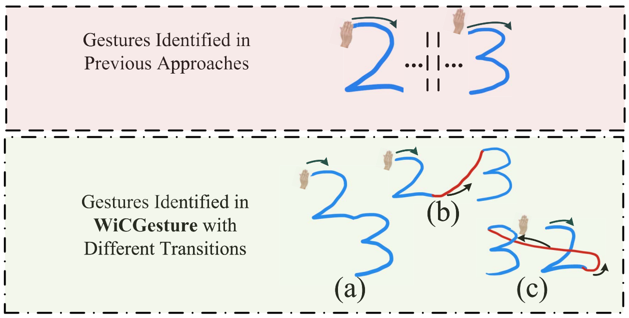


Fig. 1. Comparison between existing Wi-Fi-based gesture recognition technique (Top) and our proposed WiCGesture (Bottom). Existing techniques require the individual to pause between gestures. In contrast, WiCGesture does not has such constraint and supports continuous gesture recognition.

gesture actions from a continuous stream is not a necessary condition. Instead, the segmented clip needs merely to approximate the core information of the gesture, ensuring reliable recognition. Consequently, we have developed a segmentation method capable of extracting the pertinent information of valid gestures from a continuous stream. The essence of our method is to segment the continuous hand movement stream into basic fractions based on inherent geometric characteristics (derived from displacement and direction information) of hand movements, rather than determining the boundary at the gesture-transition level. These basic fractions, easily extracted from raw Wi-Fi signals, form a bijective representation of continuous hand-moving patterns, converting the complex motion stream into a temporal combination of basic fractions that geometrically depict hand movement. Thus, a gesture can be identified from the continuous hand movement stream by finding its bijective sequence matching its exclusive moving pattern. Thanks to the fine temporal granularity of meta motions, discrepancies between sequence boundaries and actual gesture actions do not impede the inclusion of the gesture's essential information within the bijective sequence. This supports the feasibility of continuous gesture recognition in academic contexts.

To realize it, we first analyze fundamental factors reflecting the spatial characteristics of gestures and distinguishing one gesture from others and transitions. We exploit displacement and direction variations to geometrically depict hand movement and analyze basic fragments, called *meta motion*, within each gesture. Each meta motion type has a unique geometrical description. A large number of gestures can be uniquely characterized by the types, numbers, and orders of meta motions they contain, forming a bijection with the combination of meta motions to these predefined[1] gestures.

We carefully design two different meta-motions to make them easy to extract and effectively depict gestures. Instead of directly looking for the position where each gesture starts and ends in the continuous motion streams, we decompose the complicated hand movements into a temporal sequence of meta motions and search for the subsequence which corresponds to the meta motions contained in the specific gesture. The meta-motion-based scheme can be a useful way to address the continuous gesture recognition issue. Meanwhile, a set of techniques are proposed to effectively sense hand movements from raw Wi-Fi signals, and accurately extract each meta motions with variations of the same gesture. To verify the approach, we implement a prototype system, namely, *WiCGesture*, to continuously recognize in-air gestures with two different gesture sets. Table I provides a comparison between *WiCGesture* and state-of-the-arts on a number of important dimensions.

Our main contributions are summarized as follows:

1) We introduce a novel feature, *meta motion*, to geometrically represent complex hand movements in continuous gesture scenarios.
2) We develop techniques to capture fine-grained continuous gesture motions from Wi-Fi signals and accurately extract each meta motion for different users and moving speeds.
3) We propose a meta motion-based scheme for recognizing naturally performed continuous gestures. The complex hand movement streams are converted into a sequence of meta motions, and gestures are identified using a back-tracking search-based algorithm. To our knowledge, this is the first attempt at Wi-Fi-based continuous gesture recognition.
4) We build a prototype system, *WiCGesture*, on commodity Wi-Fi devices. Our evaluation results demonstrate that *WiCGesture* effectively recognizes continuous gestures from two different sets across various users and environments, outperforming existing state-of-the-art methods.

II. RELATED WORK

Existing works related to our approach can be divided into two categories: 1) Wi-Fi-based and 2) non-Wi-Fi-based approaches. We provide details below.

[1]Gestures are performed with *predefined trajectory and order of strokes*.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

A. Wi-Fi-Based Gesture Recognition

Currently, no Wi-Fi-based approaches are specifically designed for continuous gesture scenarios. We present some representative works in the field. Existing *WiGest* [19] gesture recognition approaches can be further categorized into pattern-based and model-based approaches.

Pattern-based approaches explore the mapping relationship between hand movements and signal patterns for gesture recognition. For instance, *WiGest* [19] extracts three basic RSSI variation patterns from Wi-Fi signals corresponding to three basic hand motions and recognizes complex gestures comprising these basic motions. *WiMu* [13] uses short-time Fourier transform (STFT) to extract frequency features and generates virtual samples for multiuser gesture recognition. *Wi-Finger* [20] calculates the main component of Wi-Fi-CSI variations using principal component analysis (PCA) and generates a classifier to recognize nine different gestures. *CrossGR* [21] applies a deep-learning-based pipeline to extract user-specific and environment-independent patterns for gesture recognition. While these approaches can effectively recognize different gestures, the input data for pattern extraction is limited to single gestures, requiring extensive labor for data segmentation in continuous gesture scenarios.

Model-based approaches establish physical models between hand movements and signals to extract explicable features for gesture recognition. *WiGesture* [7] incorporates a unique position-independent feature depicting the relative direction of the hand for gesture recognition. *HandGest* [10] introduces a hierarchical sensing framework using dynamic phase vector (DPV) and motion rotation variable (MRV) for recognition. *DPSense* [8] improves the performance of existing gesture recognition applications by alleviating ambient noise and extracting a fine-grained sequence of dynamic phase variations of CSI. These works require users to keep their hands still between gestures when identifying gestures in real time and cannot be adapted to scenarios where the user moves the hand during transitions.

B. Non-Wi-Fi-Based Gesture Recognition

Some non-Wi-Fi-based approaches engage with continuous gesture recognition, which can be categorized into vision-based and sensor-based approaches. Vision-based approaches, such as [22], [23], [24], and [25], leverage gesture image information and use deep learning methods to train neural networks for continuous gesture recognition. Elmezain et al. [26] built a hidden Markov model (HMM) encoding hand motion streams for continuous gesture recognition based on hand trajectories. Jian et al. [27] developed an LSTM neural network-based scheme, using hand movement images to predict the end of each gesture for segmentation and recognition from the continuous stream. However, these machine learning-based techniques require a large amount of data samples, and their generalities are not guaranteed. Moreover, compared to Wi-Fi-based approaches, vision-based methods are sensitive to lighting conditions and may pose privacy concerns for some users. Sensor-based approaches, such as [28], utilize accelerometer, and gyroscope sensors

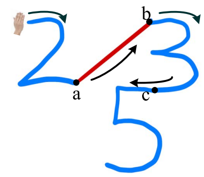


Fig. 2. Performing gestures “2–3–5.” Transitions are marked in red.

to encode hand movements for continuous gesture recognition. *ViFin* [29] extracts heterogeneous data from commodity smartwatches and applies a transfer learning-based scheme for continuous gesture recognition. While promising, these approaches require users to wear additional devices, which may be unnatural and burdensome compared to Wi-Fi-based approaches.

III. Mechanism of Continuous Gesture Recognition With Meta Motion

This section addresses the core challenge in continuous gesture recognition and introduces our solution. We employ a novel feature called meta motion to differentiate between gestures and transitions. We describe our continuous gesture recognition process with meta motion and outline how we harness its distinctive properties.

A. Challenges and Basic Idea of WiCGesture

Recognizing continuous gestures through Wi-Fi signals poses challenges, especially concerning uncertain transitions between gestures, as explained in Section I. Segmenting gestures from a continuous stream can be challenging due to the uncertainty of boundary detection. Our objective is to identify user gestures without needing precise timing or boundaries for each gesture.

Instead of extracting features at the gesture-transition level, we utilize finer-grained features to segment the stream into basic fractions based on inherent geometric characteristics of hand movements. This approach allows us to represent gesture-specific movement patterns and differentiate them from transitions in a continuous stream by identifying combinations of consecutive fractions that match gestures. Our approach is highly adaptable to a wide range of gestures, as it does not necessitate precise boundary detection.

B. Understanding the Meta Motion

We introduce *meta motion*, a fine-grained feature capturing fundamental hand movements with distinctive physical traits. This concept builds upon the previous section’s basic idea. The simplicity of these movements makes meta motion easy to extract from raw signals, enabling us to effectively characterize the unique patterns of each gesture.

To grasp the concept of meta motion, consider that a complex continuous hand movement can be geometrically represented as the user executing lines and curves. For example, continuously performing “2–3–5,” as illustrated in Fig. 2, involves consecutively executing five lines and four curves. The unique combinations of lines and curves provide

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

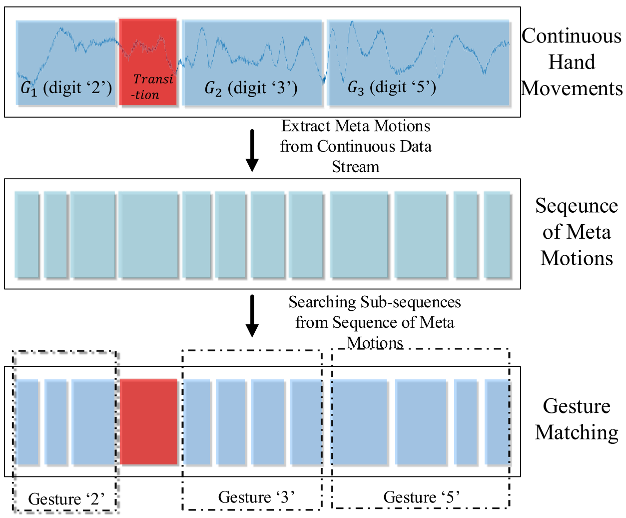


Fig. 3. Illustration of recognizing "2–3–5" with meta motions.

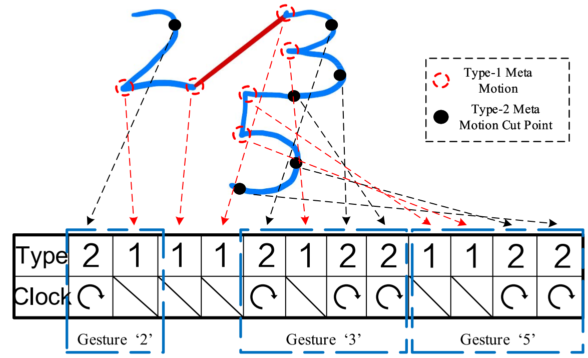


Fig. 4. Meta motion sequences of performing "2–3–5."

geometric instructions on how a gesture should be executed. This means that users, even without specific cultural or educational backgrounds, can reproduce digits by following the order of lines and curves.

Excluding the initial motion stream segment, each line or curve demonstrates sudden direction changes at its start or end. Identifying these abrupt changes allows us to infer the number and approximate locations of lines and curves, which form basic hand movement patterns. Unlike lines, curves involve continuous changes in hand direction. The rotation of curve fragments cut by abrupt direction changes distinguishes curves from lines. We represent rotation information as continuous changes in direction and displacement, but precise values are unnecessary. Instead, we partition each rotated fragment into subfragments with fixed angles and displacements, forming another basic hand movement pattern that allows us to infer the rotation of curves. Based on this analysis, we define two types of meta motions.

1) *Type 1 Meta Motion:* Hand movement with abrupt direction changes.
2) *Type 2 Meta Motion:* Hand movement that gradually rotates in the air with fixed displacement (d_k) and direction (θ_k) changes.

Note that the temporal order of different meta motions is essential to characterize hand movements without ambiguity. By obtaining a temporal sequence of meta motions, we establish a bijection to the continuous stream of in-air hand movements.

C. Feasibility of Using Meta Motions

The sequences of meta motions can distinguish different in-air gestures with unique geometrical moving patterns, including transitions. These gestures have distinct meta-motion sequences due to their unique geometric trajectories.[2] Although a simple feature, meta motion effectively captures spatial information of gestures at a coarse level, making it feasible for continuous gesture recognition.

Notably, meta motion can achieve high accuracy in recognizing single-performed gestures, as shown in Section V-D1, because it can form bijections for different gestures.

D. Identifying Continuous Gestures Using Meta Motion

Fig. 2 illustrates the basic process of identifying continuous gestures with a specific case of the user performing the "2–3–5" gestures with a transitional segment. We exclude the static segments (where the hand is still), as they can be easily identified using previous work [7], [8].

To generate the sequence S of meta motions of continuously performed gestures, we combine consecutive meta motions (regraded as S) within the continuous stream. We then identify the gestures by finding subsequences in S that match with the reference sequence. Fig. 3 shows an example of recognizing a gesture by matching the subsequence in S with the reference.[3] Fig. 4 shows the specific meta motions for continuous writing of 2–3–5." The type-1 meta motions are marked with red circles. We mark the end of corresponding type-2 meta motions with black dots every time we estimate the displacement and direction change d_k and θ_k. The subtable in Fig. 4 presents the corresponding sequence of meta motions, which enables us to distinguish between different gestures and transitions.

The key advantage of our approach is that it can adapt to continuous gestures freely performed by users, without making assumptions about how transitions are performed. However, to ensure that the complete gesture can always be found, we require that the beginning or end of a meta-motion correspond to the beginning and end of gestures. We satisfy this requirement by analyzing the gesture as follows.

Our approach's significant advantage lies in its adaptability to user-performed continuous gestures, without presuming specific transition mechanisms. Nonetheless, for complete gesture detection, we mandate that the start or end of a meta-motion corresponds to the start and end of gestures. This requirement is satisfied as follows. We first consider the case when the hand moves from gestures to transitions. When the hand enters the transition segment (Position a in Fig. 2) from the gesture G_1, the moving direction will certainly change.

[2]Although carefully explored the meta motion-based scheme for the digit set, we also applied the scheme to another set to verify its generality.

[3]More details are presented in Section IV-E.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

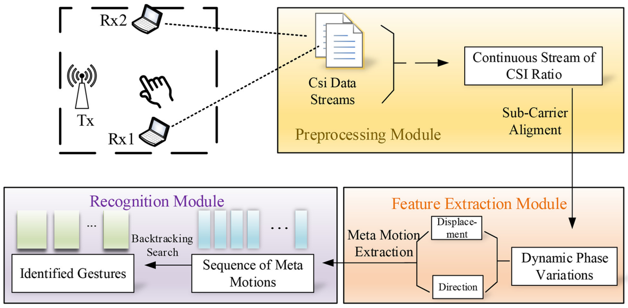


Fig. 5. Overview of WiCGesture system design.

Otherwise, the hand movement can still be considered as an extension of the previous gesture. The change in angular tendency introduces a new meta motion based on our previous definitions.

Similarly, when the hand enters a gesture from the transition (Position b in Fig. 2), a changed angular tendency will introduce a new meta motion. Otherwise, the transition can be considered as the beginning part of the following gesture.

Finally, when the user directly performs a new gesture G_3 after finishing the old one G_2 without any transition (Position c in Fig. 2), the angular tendency will soon change at the new gesture. It is possible that part of the beginning of the new gesture can be wrongly identified as the tail of the previous gesture. However, this does not affect the meta motion sequence of G_3, and G_3 can still be correctly recognized.

IV. WiCGesture Design and Implementation

In this section, we present the design and implementation of *WiCGesture* for recognizing continuous gestures.

A. Practical Implementation Challenges

Recognizing different continuous gestures accurately with commodity Wi-Fi devices requires overcoming several practical challenges in system implementation, including:

How to Accurately Sense Continuous Hand Movements From Raw Wi-Fi Signals: In practice, Wi-Fi signals are prone to hardware and environmental noise, and the speed of hand movements can vary when performing gestures continuously. A limited sampling rate of Wi-Fi signals may fail to capture fast hand movements flawlessly. Therefore, obtaining a high-quality base signal from the raw Wi-Fi signal is essential.

How to Extract Meta Motions From Continuous Data Streams: To accurately calculate the physical parameters (displacement and direction changes) for meta motion extraction, information, such as the initial hand location and transceiver positions are required [7], [16], [30]. Yet, acquiring this additional data can be cumbersome and unnatural. Hence, extracting meta motions from continuous data streams in a natural manner remains a challenge.

B. Overview of WiCGesture

As shown in Fig. 5, *WiCGesture* has three components: 1) *preprocessing*; 2) *feature extraction*; and 3) *recognition* modules. Preprocessing module removes noise from raw CSI signals and uses multicarriers to capture hand movements addressing the first challenge. Feature extraction module addresses the second challenge by extracting meta motions without prior knowledge of hand or transceiver locations. Recognition module identifies subsequences of gestures and addresses the subgesture issue to ensure accurate identification.

C. Preprocessing of Continuous Data Stream From Raw Wi-Fi Signals

Wi-Fi signals propagate over multiple paths from the transmitter to receiver, including the Line-of-Sight (LoS) path and other reflection paths in the environment. The moving hand will introduce a new changing reflection path resulting in the variation of CSI on the receiver. Specifically, the received CSI at time t with frequency f and wavelength λ can be modeled as follows [31] and [32]:

$$H(f,t) = H_s(f,t) + A(f,t)e^{-j2\pi\frac{l(t)}{\lambda}} \tag{1}$$

where $H_s(f,t)$ is the static component, $A(f,t)$, $e^{-j2\pi[l(t)/\lambda]}$, and $l(t)$ are the attenuation, phase, and length of the dynamic reflection path, respectively. The $A(f,t)e^{-j2\pi[l(t)/\lambda]}$ will rotate with the changing reflection path, which establishes the quantitative relationship between the hand movement and the dynamic phase. When l changes with the length of λ, the dynamic phase φ will accordingly change 2π. As pointed out in [8], the dynamic phase variations $\Delta\varphi$ act as an effective indicator for gesture sensing.

Therefore, the goal of the preprocessing module is to obtain a fine-grained dynamic phase sequence from the receiver as the continuous data stream. However, the CSI collected on the commodity devices suffers from random offset noise in the phase including channel frequency offset (CFO) and sample frequency offset (SFO), which impedes us from extracting dynamic phase sequence. We adopt widely used CSI ratio [7], [33] to remove the phase offsets as follows:

$$H_r(f,t) = \frac{H_1(f,t)}{H_2(f,t)} \tag{2}$$

where $H_1(f,t)$ and $H_2(f,t)$ are CSI from two antenas on the same receiver. As shown in Fig. 6, it has been proved [7], [33] that $H_r(f,t)$ can approximately recover the I–Q form of $H(f,t)$ in the complex plane, which is capable of gesture sensing.

As previously mentioned, $H_r(f,t)$ may not accurately represent rapid hand movements with a low sampling rate. For instance, with a default sampling rate of 400 Hz as in [7] and [10], if a user waves their hand at a high speed of approximately 15 cm/s, there will be fewer than 30 CSI samples to capture dynamic phase changes of π.[4] With a reduced number of samples, the characterization of hand movements becomes more susceptible to noise interference, resulting in a significant drop in system performance. Increasing the hardware-level sampling rate may not be a viable solution as it places a heavy load on the network card, and commodity cards have limitations on their maximum sampling rates [34].

[4]Length of LoS is set to 1 m, length of the reflection path is set to 1.5 m and the user moves the hand perpendicular to Fresnel zone layers.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

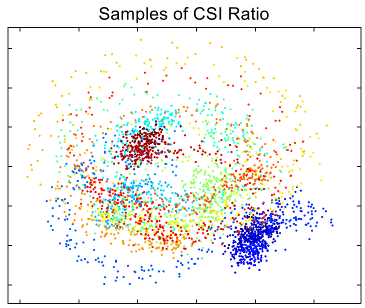


Fig. 6. CSI ratio samples in the complex plane of performing “2–3–5” (painted from blue to red).

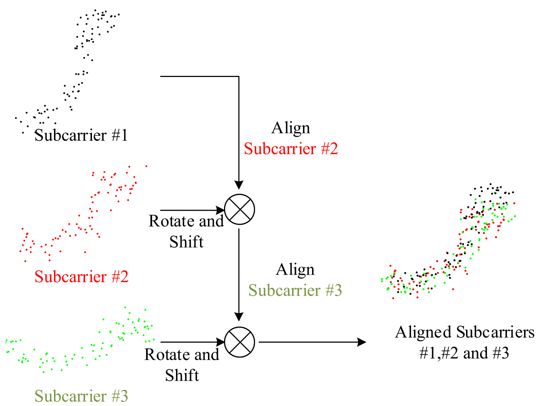


Fig. 7. Illustration of aligning multicarriers.

To solve this problem, we generate a new base signal utilizing multiple subcarriers of collected CSI to relieve the negative impact of the low sampling rate. The basic idea is to align subcarriers at the complex plane as shown in Fig. 7. Subcarriers with similar frequencies exhibit similar CSI sample distributions in the complex plane, enabling us to rotate and shift samples from one subcarrier with those from an adjacent subcarrier. This approach provides more samples for a more accurate representation of dynamic phase changes.

Specifically, using CSI-ratio [7], [33], we obtain the $\mathcal{H}_r(f_i)$ for the ith subcarrier. We set a sliding window T (0.2 s for 400 Hz) to process $\mathcal{H}_r(f_i)$ in slice. We present $\mathcal{H}_r(f_i)$ as follows:

$$\mathcal{H}_r(f_i) = \{\mathcal{H}_r(f_i, t_0), \mathcal{H}_r(f_i, t_0 + T) \cdots\}. \quad (3)$$

For nth slice $\mathcal{H}_r(f_i, t_0 + nT)$, we rotate and shift the corresponding nth slice $\mathcal{H}_r(f_{i+1}, t_0 + nT)$ at $i + 1$ th subcarrier as follows:

$$\mathcal{H}_{r,\mathrm{rotate}}(f_{i+1}, t_0 + nT) = e^{-j\theta}\mathcal{H}_r(f_{i+1}, t_0 + nT) + v_{\mathrm{shift}} \quad (4)$$

where θ and v_{shift} are the rotation angle and shift vector, respectively. We determine these two factors using genetic algorithm when the variance of $\mathcal{H}_r(f_i, t_0 + nT) - \mathcal{H}_{r,\mathrm{rotate}}(f_{i+1}, t_0 + nT)$ reaches the minimum. For each sample of $\mathcal{H}_r(f_i, t_0 + nT)$ and $\mathcal{H}_{r,\mathrm{rotate}}(f_{i+1}, t_0 + nT)$ with the same time-stamp, their geometric average is calculated as the sample in our new base signal $\mathcal{H}_r(f_{(i,i+1)}, t_0 + nT)$. Thus, the generated new base signal is presented as follows:

$$\mathcal{H}_r\big(f_{(i,i+1)}\big) = \{\mathcal{H}_r\big(f_{(i,i+1)}, t_0\big), \mathcal{H}_r\big(f_{(i,i+1)}, t_0 + nT\big) \cdots\}. \quad (5)$$

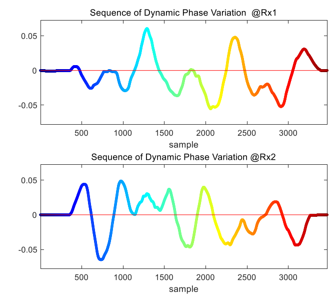


Fig. 8. Dynamic phase variations on two receivers (painted from blue to red).

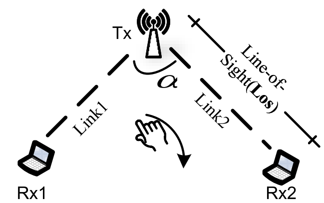


Fig. 9. Sensing hand movements with two pairs of transceivers.

We conduct an iterative process to align more subcarriers [e.g, aligning $\mathcal{H}_r(f_{i+2})$ on $\mathcal{H}_r(f_{(i,i+1)})$]. In practice, we align three subcarriers at the balance of the computation cost and performance. We also apply the framework in DPSense [8] to mitigate the influence of ambient noise, select high-quality subcarriers, and extract fine-grained dynamic phase variations. It utilizes a PCA-based approach to extract raw dynamic phase variations. These variations are then processed separately based on signal quality. Low-quality signals are discarded, and their dynamic phase variations are estimated using prior knowledge. The final extracted sequence of dynamic phase variations $\Delta\varphi$ as shown in Fig. 8 acts as the continuous data stream in *WiCGesture* for further recognition.

D. Meta Motion Extraction From Continuous Data Stream

Revisiting the definition of meta motions in Section III-B, we need to estimate the displacement d and direction change $\Delta\theta$ of hand movements for feature extraction. As mentioned before, direct calculation of d and $\Delta\theta$ is infeasible as the initial location of the hand and locations of transceivers are unknown [7]. Luckily, it is unnecessary to require the precise value of d and $\Delta\theta$. As long as the estimated $\hat{d}$ and $\hat{\Delta\theta}$ are positively correlated with d and $\Delta\theta$, the extracted meta motions can still replace the ideal meta motion to depict continuous hand movements. We calculate $\hat{d}$ and $\hat{\Delta\theta}$ using continuous data streams ($\Delta\varphi$) from two receivers.

1) Estimate Displacement: It is difficult to utilize only one pair of transceivers to estimate the 2-D displacement of in-air gestures performed in the plane. Therefore, as shown in Fig. 9, we place two pairs of transceivers along different sides of the hand. Two sequences of $\Delta\varphi_1$ and $\Delta\varphi_2$ are derived from two receivers (Rx1 and Rx2). As previously illustrated, $\Delta\varphi$ is introduced by the reflection path changes Δl, which is also related to Doppler speed v_{doppler} of the moving hand [30]. Considering the direction remains constant in a short time window, we instinctively note that the greater

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

d is, the larger Doppler velocity of $v_{1,\text{doppler}}$ and $v_{2,\text{doppler}}$ will be. Thus, we only leverage two Doppler to estimate $\hat{d}$. When the angle α of two links shown in Fig. 9 is approximately set to $(1/2)\pi$, we prove that Δd monotonically increase with the $\sqrt{v_{1,\text{doppler}}^2 + v_{2,\text{doppler}}^2}$. The mathematical derivations are presented in the Appendix. Thus, the estimation of displacement over the time T can be expressed as follows:

$$\hat{d}(T) = \int_0^T \sqrt{v_{1,\text{doppler}}^2 + v_{2,\text{doppler}}^2} dt. \quad (6)$$

In this case, the $\Delta\varphi_1$ and $\Delta\varphi_2$ are only needed for estimation. Therefore, without requiring extra information, we calculate $\hat{d}$ as the estimation of displacement for meta motion extraction.

2) Estimate Direction Variations: We propose a new method for estimating direction variations in hand gestures that is more stable across different hand speeds and reduces abrupt changes caused by hand shivers. Our approach builds upon the motion navigation primitive (MNP) developed inWiGesture [7], which reflects the patterns of direction variations for gesture recognition. However, MNP has some limitations: it is sensitive to different hand speeds, resulting in varying direction changes for the same trajectory. Additionally, the hand's instability during continuous gesture performance can introduce abrupt changes in MNP.

To address the limitations of MNP, we extract the pattern of direction variations across displacement instead of time, as this is more consistent for the same type of gesture with similar trajectories, regardless of the hand speed. The intuition behind this is that for the same type of gesture with similar trajectories, the direction variations across the displacement are also similar, no matter what the moving speed is.

We use a sliding window approach to calculate the average value of MNP for each fragment of a fixed displacement Δd, and convert the sequence of MNP average values into a sequence of estimated direction variations across displacement, denoted as M_{all}. The Δd, can be inferred in the previous section and is empirically set to 0.5 cm.

The estimation of direction changes $\Delta\hat{\theta}$ is then calculated as the change in M_{all} as follows:

$$\Delta\hat{\theta} = \Delta M_{\text{all}}. \quad (7)$$

This method is less susceptible to the effects of hand shivers and can be used across different hand speeds. Therefore, it improves upon the previous approach by providing a more stable and accurate estimation of direction variations in hand gestures.

3) Identify Meta Motions: Our approach identifies two types of meta motions ($\hat{d}$ and $\Delta\hat{\theta}$) to capture hand movements. The first type represents abrupt direction variations, as shown in Fig. 10. We detect these locations by thresholding $\Delta\hat{\theta}$. The second type corresponds to a certain change in both displacement and direction. We detect these by dividing hand movements into fragments and calculating the cumulative changes in displacement and direction (Δd and $\Delta\theta$) from the start of each fragment. A type-2 meta motion is detected when both Δd and $\Delta\theta$ exceed empirically set thresholds (d_k and θ_k). We set d_k to 1 cm based on typical handwriting gesture sizes and θ_k to $(1/2)\pi$ to ensure consistent meta motions for the corresponding arcs of the same kind of gestures.

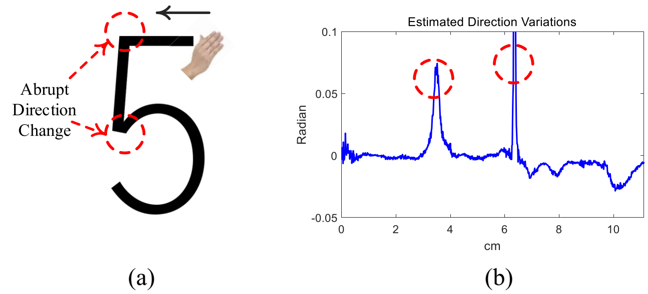


Fig. 10. Identify abrupt direction change of '5'. (a) Writing digit '5'. (b) Estimated direction variations of '5'.

To determine clock direction, we use the increasing/decreasing trend of $\Delta\hat{\theta}$, which corresponds to the counter-clockwise/clockwise direction of hand movements. The resulting sequence of meta motions ($\mathcal{S}_{\text{meta}}$) is continuously updated in real time as the user performs gestures, which is shown as follows:

$$\mathcal{S}_{\text{meta}} = \{s_0, s_1, \ldots, s_n\} \quad (8)$$

where each s_i represents a meta motion.

E. Continuous Gesture Recognition With Meta Motion Using Backtracking Searching Algorithm

With extracted $\mathcal{S}_{\text{meta}}$, we design a backtracking algorithm to recognize gestures. Here, are the specifics:

1) Generate Reference Sequence: To recognize predefined gestures, we generate a reference sequence $\mathcal{S}ref$ for each gesture. A subsequence in $\mathcal{S}meta$ matching $\mathcal{S}_{\text{ref}}$ indicates a recognized gesture. We obtain the reference sequence by having the user perform each gesture and extracting the corresponding meta motion sequence. This process requires only a few precollected samples and is performed once for each gesture set.

2) Recognize Gesture With Sequence Matching: To recognize gestures, the system first detects whether the hand is moving or static using the DPSense method [8]. When the hand is moving, we continuously extract meta motions and append them to $\mathcal{S}_{\text{meta}}$. Once a meta motion is identified, we create a subsequence $\mathcal{V}(i, q)$ as follows:

$$\mathcal{V}(i, q) = \{s_i \ldots s_q\} \text{ where } (p < i < q) \quad (9)$$

where s_q is the currently identified meta motion.

We match each $\mathcal{V}(i, q)$ with $\mathcal{S}ref$ of all gestures. When $\mathcal{V}(i, q)$ and $\mathcal{S}ref$ have the same meta motions with the same order, we successfully find the subsequence $\mathcal{V}(i, q)$ that matches with a reference. However, it is possible that $\mathcal{V}(i, q)$ corresponds to a subset of a bigger gesture. For example, digit "2" can be the prefix of digit "3" To avoid misidentifying, we test whether $\mathcal{V}(i, q)$ is a prefix or suffix of another gesture. To test the suffix, we look back m meta motions before s_i, where m increases from 1 to a predefined maximum value. If $\mathcal{V}(i - m, q)$ matches with another reference sequence, it will be recognized as the corresponding gesture. To test the prefix,

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

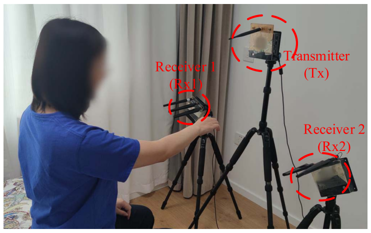
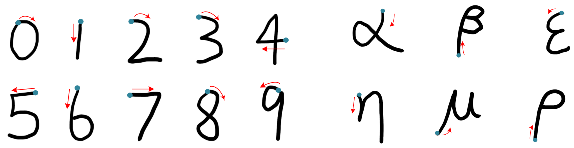


Fig. 11. Deployment of WiCGesture system.



(a) (b)

Fig. 12. Illustration of two evaluated gesture sets. (a) Gesture set S_1. (b) Gesture set S_2.

we wait for more n meta motions after s_q. The gesture will be recognized if $\mathcal{V}(i, q + n)$ matches with another reference sequence. The maximum values of m and n are set to 2, which are sufficient to detect all subset situations in practice. If no suffix or prefix is detected within the maximum of m and n, a gesture is recognized if its reference matches with $\mathcal{V}(i, q)$.

Our algorithm can recognize a wide range of complicated gestures in the continuous movement stream, except for digits 1 and 7, which are simple and may be confused with transitions. However, we have basic rules to easily identify them. For digit 1, we detect its direction using extracted $\hat{\theta}$ and recognize it if the hand moves downward between two type-1 meta motions and it is not the suffix or prefix of other gestures. For digit 7, we also detect its direction using $\hat{\theta}$ and recognize it if the hand moves to the right before the type-1 meta motion, and it is not the suffix or prefix of other gestures.

V. Evaluation

In this section, we conduct comprehensive experiments to evaluate the effectiveness of *WiCGesture*.

A. Evaluation Setup

1) System Deployment: As shown in Fig. 11, we employ three mini-PCs configured as one transmitter (Tx) and two receivers (Rx), mounted on tripods to form a vertical plane for improved in-air hand movement sensing. Each device features a commodity Intel 5300 network card and external omni-directional antennas for Wi-Fi signal transmission and CSI collection. The *WiCGesture* system operates on a laptop which continuously receives CSI data streams from both Rxs, enabling real-time gesture recognition.

2) Data Collection: To assess *WiCGesture*'s performance, we employ two gesture sets: digit set S_1 and mathematical symbol set S_2, comprising six Greek letters. Fig. 12(a) and (b) display S_1 and S_2, respectively. We devise multiple gesture combinations for *WiCGesture* to evaluate on both sets. Each data point represents a gesture combination. Specifically, S_1 is assessed using four triples, three quadruples, and two quintuples, while S_2 is evaluated with three quadruples. See Tables II and III for details.

TABLE II
Gesture Combinations of S_1

Combinations	Descriptions
Triple	{2, 3, 5} {0, 4, 6} {7, 8, 9} {1, 4, 9}
Quadruple	{0, 3, 5, 7} {3, 5, 6, 8} {1, 2, 4, 9}
Quintuple	{0, 2, 4, 6, 8} {1, 3, 5, 7, 9}

TABLE III
Gesture Combination of S_2

Combinations	Descriptions
Quadruple	$\{\alpha, \beta, \epsilon, \eta\}$ $\{\mu, \rho, \alpha, \beta\}$ $\{\epsilon, \eta, \mu, \rho\}$

TABLE IV
Default Parameters in Experiments

Description	Setting	Description	Setting
Gesture Sets	S_1	Moving Speeds	8 cm/s
Environments	Office Room	Lengths of LoS	80 cm
Users	U_1	Sampling Rates	400 Hz

Users participate in experiments conducted in three distinct indoor settings: an office, a living room, and a bedroom. Each user performs every gesture combination ten times in each room while seated in front of transceivers, extending their hands into the plane formed by two links (Tx-Rx1 and Tx-Rx2) and executing in-air gestures.

Users follow trajectories to perform gestures, but transitions remain unrestricted. They may pause between gestures or move their hands freely during transitions, completing combinations in their preferred manner, as long as the predefined combination is executed.

B. Overall Performance of WiCGesture

We present recognition results with default parameters in Table IV. Fig. 13 displays the confusion matrix for gesture set S_1. *WiCGesture* achieves an average accuracy of 89.6% for digit recognition in continuous scenarios. Occasionally, digit "3" is misidentified as "2," since "2" is a subset of "3." Digit "8" has slightly lower accuracy due to its complexity and additional meta motions. The confusion matrix for S_2 is in Fig. 14. Despite Greek letters being more complex than digits, *WiCGesture* recognizes them with a high average accuracy of 88.3%.

C. System Robustness

We assess *WiCGesture*'s robustness under various factors using gesture combinations from digit set S_1, given its relevance to HCI-based smart applications.

1) Impact of Users: We examine performance across 18 users (aged 21–44) from the university with the default parameters, including 6 females (Users 3–5 and 12–14) and 12 males. Users 14–18 are all over 30 years old. Four (Users 1–4) are co-authors. The result is shown in Fig. 16. The

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

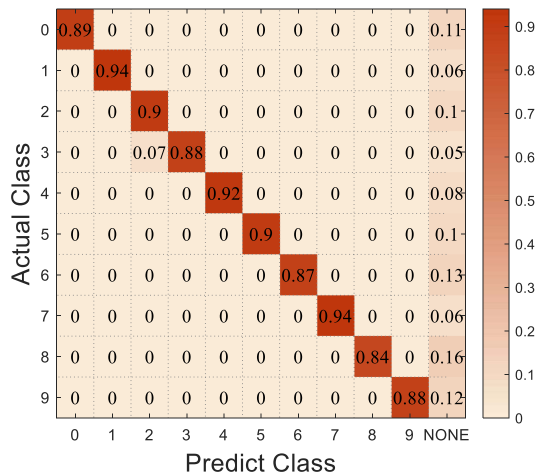


Fig. 13. Confusion matrix of S_1.

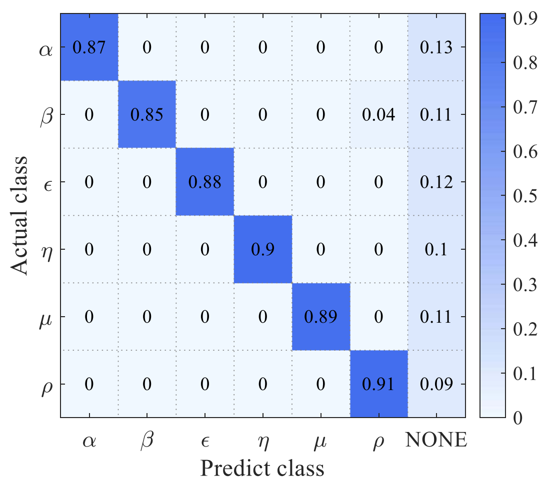


Fig. 14. Confusion matrix of S_2.

first four users have slightly higher average accuracy (92.7%), possibly due to familiarity with the system and stable hand movements. Nonetheless, *WiCGesture* effectively recognizes gestures of other users (Users 5–18) with an average accuracy of 88.7%. We observe that the gender and age factors do not significantly affect the outcome. We note that User 12's results are slightly lower (82%). One possible explanation is that her hand's relatively small size, and consequently, the smaller reflecting surface, leads to weaker reflection signals.

2) Impact of Speeds: We examine the influence of varying hand movement speeds on system performance. Users are instructed to move their hands at approximately 4, 8, 15, and 20 cm/s. Fig. 17 shows the average recognition accuracy at different speeds. Thanks to our preprocessing techniques in Section IV-C, *WiCGesture* adapts to diverse speeds. However, the accuracy at 20 cm/s is slightly lower, as rapid hand movements may lead to less accurate digit execution.

3) Impact of Environments: We assess the effect of different environments on system performance. Fig. 15 shows three indoor spaces: 1) an office; 2) a living room; and 3) a bedroom, furnished with typical daily life items. Fig. 18 displays the results, indicating no significant accuracy difference across environments, demonstrating *WiCGesture*'s robustness.

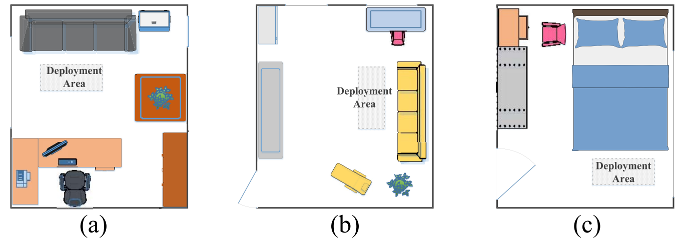


Fig. 15. Layouts of three indoor environments. (a) Office room. (b) Living room. (c) Bed room.

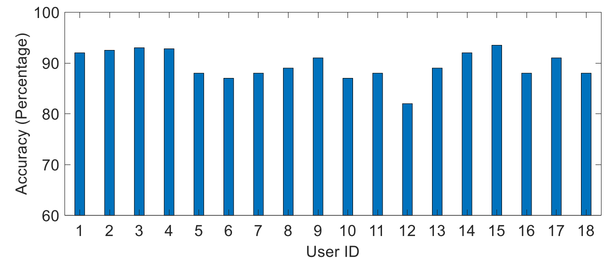


Fig. 16. Impact of different users on WiCGesture performance.

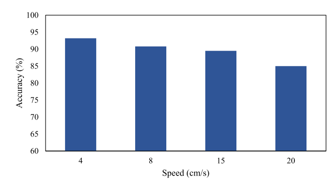


Fig. 17. Different speeds.

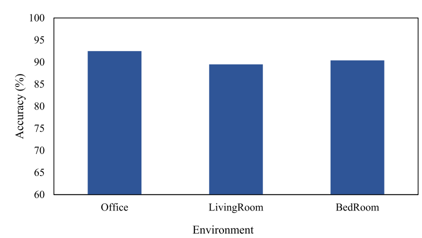


Fig. 18. Different environments.

4) Impact of Device Deployments: We assess the influence of device deployments on system performance. As indicated in Section IV-D, the extraction of meta-motion is based on the *WiGesture* [7], which shares a common feature: transceivers do not need to be deployed with exact precision or predetermined positioning, provided that two links are not aligned in parallel. Our evaluation focuses on two specific deployment factors as shown in Fig. 9: 1) LoS Length and 2) Link Angles(α).

LoS Length: We evaluate how different LoS lengths of two links affect performance. In addition to the default 80 cm LoS, we test 100, 120, and 150 cm. Fig. 21 shows that recognition accuracy decreases with increasing LoS distance due to weaker signal propagation.

Angles of Two Links: We assess the impact of different angles α between two links (Tx-Rx1 and Tx-Rx2) on system performance. In addition to the default angle of $(1/2)\pi$, we conducted tests with angles of $(1/3)\pi$, $(2/3)\pi$, and $(5/6)\pi$. Fig. 19 illustrates the accuracy, indicating that *WiCGesture* achieves consistent results. It is worth noting that performance

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

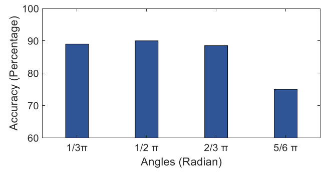


Fig. 19. Different angels.

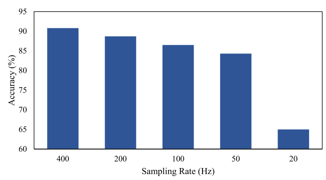


Fig. 20. Different sampling rates.

decreases under an angle of $(5/6)\pi$. This reduction can be attributed to the near parallel alignment of the two links, potentially leading to some misidentification of meta-motions.

Note that the system exhibits increased sensitivity on a plane perpendicular to antennas due to antenna polarization factors. Consequently, during the sensing process, users execute continuous gestures within this plane.

5) Impact of Sampling Rates: We examine the effect of varying CSI record sampling rates on performance. We test *WiCGesture* at four lower rates: 1) 200; 2) 100; 3) 50; and 4) 20 Hz, with the default rate at 400 Hz. In the setup, we analyze the electromagnetic environment using dedicated software and configure the system to work on channel 149 in the 5G frequency range to avoid interference from other WiFi channels. Fig. 20 presents the results. *WiCGesture* maintains relatively high accuracy (85%) at 50 Hz. Thanks to the alignment with multiple carriers that exhibit higher signal quality as discussed in Section IV-C, and with a sampling rate of 50 Hz, we can acquire 150 data points per second when aligning with three subcarriers. This configuration continues to support the effective recognition of gestures by the system. However, we observe that accuracy drops significantly at 20 Hz due to insufficient CSI samples for capturing hand movements at the default speed (8 cm/s). In complex settings with multiple WiFi channels in the room, our system becomes more susceptible to interference. If there is a conflict between our sensing channel and other communication channels, it may result in significant issues, such as packet loss or the inability to receive CSI.

D. Comparison With State-of-Arts

We assess *WiCGesture* against four state-of-the-art systems, namely, *Wi-Finger* [9], *WiGesture* [7], *HandGest* [10], and *DPSense* [8]. We evaluate gesture combinations from S_1 across these systems in two distinct scenarios. While not designed for continuous gesture recognition, the latter three systems can recognize multiple gestures in real time with the user's hand stationary during transitions.

1) Comparison of Recognizing Single Gestures: We compare single-gesture recognition performance to validate the effectiveness of meta motion in characterizing gesture

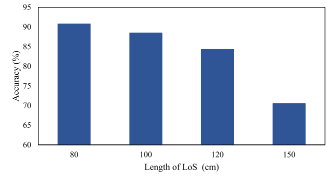


Fig. 21. Different LoS.

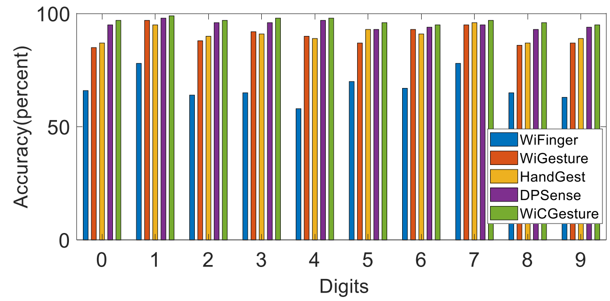


Fig. 22. Comparison of recognizing single gestures.

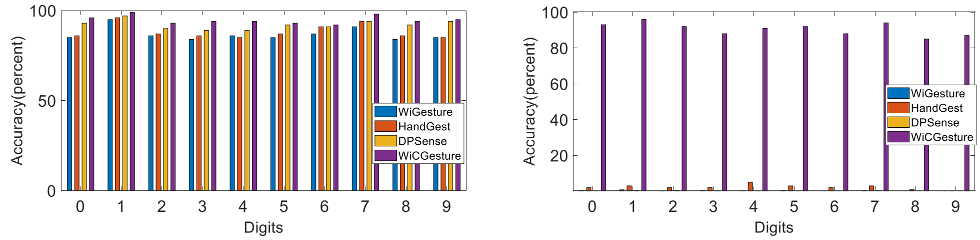
movements. Fig. 22 shows the results. *WiCGesture* outperforms *Wi-Finger* [9], *WiGesture* [7], *HandGest* [10], and *DPSense* [8] by 29.4%, 6.8%, 6%, and 1.7%, respectively. *Wi-Finger* [9] has the lowest accuracy, as it relies on a pattern-based recognition scheme. However, signal patterns vary with location and orientation for the same gesture [7], [35]. Since users are not required to perform gestures in a specific position, *Wi-Finger* [9] cannot ensure high accuracy. Though the other three systems have high accuracy, *WiCGesture* performs slightly better due to our proposed techniques in Section IV-D2, which leverage *WiGesture* [7] and *DPSense* [7] while adapting to different moving speeds.

2) Comparison of Recognizing Continuous Gestures: We compare continuous gesture recognition using the combinations from the digit set S_1 shown in Table II. *Wi-Finger* [9], a pattern-based approach, can only handle single-gesture input data, making it unsuitable for continuous gestures. Thus, we compare *WiCGesture* with *WiGesture* [7], *HandGest* [10], and *DPSense* [8]. In particular, we perform the comparison in two scenarios. The results are shown in Figs. 23.

a) Scenario A (Continuous gestures with paused transitions): In this scenario, users pause for at least one second between consecutive gestures. *WiGesture* [7], *HandGest* [10], and *DPSense* [8] can recognize continuous gestures with paused transitions. Section V-D1 shows the recognition accuracy of the four systems. All perform well, but *WiCGesture* slightly outperforms the others by 8%, 6.5%, and 2.7%, achieving an average accuracy of 94.8%. The meta motion's ability to better characterize hand movements contributes to this advantage.

b) Scenario B (Continuous gestures with moving transitions): In this scenario, users' hands move continuously between gestures. Section V-D1 shows the recognition accuracy of the four systems. *WiGesture* [7], *HandGest* [10], and *DPSense* [8] fail to identify gesture combinations with moving transitions, recognizing them as a single undefined gesture. In contrast, *WiCGesture* effectively recognizes continuous gestures in this scenario, significantly outperforming other

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 23. Comparison of recognizing continuous gestures. (a) Scenario A: Continuous gestures with paused transitions. (b) Scenario B: Continuous gestures with moving transitions.

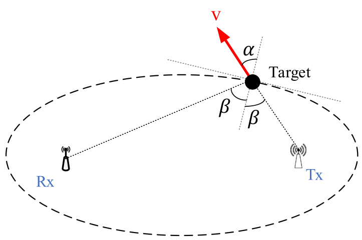


Fig. 24. Relationship of doppler velocity and real velocity with one link.

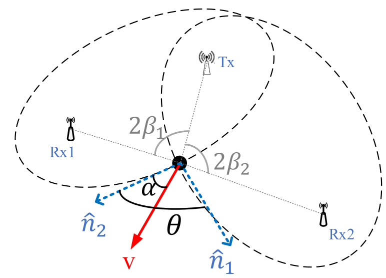


Fig. 25. Relationship of doppler velocities and real velocity with two pairs of links.

systems with an average accuracy of 90.6%. The results demonstrate the superiority of *WiCGesture* and what it means to break new ground in the field of Wi-Fi-based continuous activity recognition.

VI. Discussions

A. Cascading Misidentifying

Cascading misidentification occurs when incorrectly recognizing a gesture impacts the recognition of subsequent gestures. This can arise if the start of the next gesture is wrongly considered part of the previous one. Although rare and difficult to eliminate in nonlearning-based approaches like *WiCGesture*, cascading misidentification usually affects less than two following gestures. If the second gesture is influenced by the first, it is likely identified as a transition, not impacting the third gesture's recognition.

We have outlined a preliminary approach: periodically reevaluating previously recognized action segments to identify and correct potential cascading errors. In practical terms, let's consider a user's two continuously performed gestures, labeled as a_1 and a_2. If an inaccurate meta motion recognition within a_1 in its misinterpretation as a_1', where a_1' includes a meta motion from a_2, leading to the failure to recognize a_2 and the cascade effect where the initial gesture's error impacts the second one. To tackle this, during reevaluation, we can retroactively match meta motion subsequences in reverse order, starting with a_2 before a_1. This approach ensures that a_2's recognition remains unaffected by a_1. During reevaluation, multiple outcomes may emerge. the system can present these distinct results, allowing users to make selections based on their preferences or contextual information.

This approach lays the foundation for our future system optimization to enhance overall robustness.

B. Capability of Meta Motion

Meta Motion identifies gestures with predefined performing orders. For gestures with multiple orders, multiple reference sequences can be generated to fully represent the gesture, maintaining our scheme's feasibility.

C. Real-World Applications of WiCGesture

While *WiCGesture* holds promise, we cannot guarantee to recognize all the public gestures with our single approach. For instance, it faces challenges when it comes to continuous typing of English characters due to certain pairs of characters sharing the same subsequence of meta motions (such as "a" and "d," "N" and "Z"). However, drawing inspiration from Handgest [10], it is feasible to explore the incorporation of more distinctive features to enable hierarchical gesture recognition for different character sets.

Despite the aforementioned limitations, the significance of *WiCGesture* lies in its pioneering approach to continuous gesture recognition using Wi-Fi signals. It holds the potential for seamless integration with other systems, opening the door to the development of real-life human–computer interaction HCI-based smart applications.

VII. Conclusion

In this article, we propose a novel gesture recognition approach to recognize continuous gestures. We first introduce a new feature named *meta motion* to depict hand motions. The continuous hand movement stream can be converted into a sequence of meta motions. A set of techniques are also proposed to effectively extract *meta motion* from raw signals. To identify consecutive gestures with unpredictable transitions,

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

we proposed a meta motion-based scheme to find the subsequence of each gesture from the meta motion sequence using a backtracking-based searching algorithm. We build a prototype system named *WiCGesture* on commodity devices, which is the first continuous gesture recognition system leveraging Wi-Fi signals. Extensive evaluation results show that *WiCGesture* can recognize different gestures continuously performed by users, and is robust to various real-world factors.

Appendix
Derivation of Displacement Estimation

We verify the correctness of (6) by modeling the hand as a moving object on a certain ellipse with a transmitter and receiver as the foci. We derive the relationship between Doppler velocity v_d and real velocity v as shown in the following:

$$v_d = v \cdot \cos\alpha \cdot 2\cos\beta \tag{10}$$

where α is the angle between v and outer normal direction of the ellipse shown in Fig. 24, β is the angle between the normal and lines that connect moving object and transceivers.

We obtain Doppler velocities v_{d_1} and v_{d_2} from two links and derive the equations in the following:

$$\begin{cases} v_{d_1} = v \cdot \cos\alpha \cdot 2\cos\beta_1 \\ v_{d_2} = v \cdot \cos(\theta - \alpha) \cdot 2\cos\beta_2 \end{cases} \tag{11}$$

where as shown in Fig. 25, α is the angle between v and the normal direction of ellipse TX-Rx1, and the angle between v and the normal direction of ellipse Tx-Rx2 is $\theta - \alpha$. The β_1 and β_2 are angles between normal directions of two ellipses and lines which connect moving object to devices, respectively. Combining (10) and (11), we arrive at

$$v^2 = \frac{v_{d_1}^2 \cos^2\beta_2 + v_{d_2}^2 \cos^2\beta_1 - 2v_{d_1}v_{d_2}\cos\beta_1\cos\beta_2\cos\theta}{4\sin^2\theta\cos^2\beta_1\cos^2\beta_2}. \tag{12}$$

Combining (10) and (11), we arrive at (12).

In our scenario, we assume that the performing area is relatively small compared to the two links. Consequently, the ellipses formed by the two pairs of transceivers remain nearly unchanged in a short period, and we can consider θ, β_1, and β_2 as constants. With this assumption, we have the following:

$$\begin{cases} \frac{\partial v^2}{\partial v_{d_1}} = \frac{v_{d_1}\cos\beta_2 - v_{d_2}\cos\beta_1\cos\theta}{2\sin^2\theta\cos^2\beta_1\cos\beta_2} \\ \frac{\partial v^2}{\partial v_{d_2}} = \frac{v_{d_2}\cos\beta_1 - v_{d_1}\cos\beta_2\cos\theta}{2\sin^2\theta\cos\beta_1\cos^2\beta_2}. \end{cases} \tag{13}$$

By placing transceivers with $\theta \approx (\pi/2)$ and assuming users perform gestures in the central part of the area formed by the two links, we can simplify (13) as follows:

$$\begin{cases} \frac{\partial v^2}{\mathrm{d}v_{d_1}} = v_{d_1} \\ \frac{\partial v^2}{\mathrm{d}v_{d_2}} = v_{d_2}. \end{cases} \tag{14}$$

The changing rate of v^2 with respect to v_{d_1} and v_{d_2} is the same as the changing rate of $(v_{d_1}^2 + v_{d_2}^2)$ with respect to v_{d_1} and v_{d_2}. Therefore, we have proved the correctness of (6).

References

[1] Y. Ma, G. Zhou, and S. Wang, “WiFi sensing with channel state information: A survey,” *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, 2019.

[2] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang, “Wireless sensing for human activity: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1629–1645, 3rd Quart., Aug. 2019.

[3] F. Wang, W. Gong, and J. Liu, “On spatial diversity in WiFi-based human activity recognition: A deep learning-based approach,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2035–2047, Apr. 2019.

[4] K. Niu et al., “WiMorse: A contactless morse code text input system using ambient WiFi signals,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9993–10008, Dec. 2019.

[5] H. Kang, Q. Zhang, and Q. Huang, “Context-aware wireless-based cross-domain gesture recognition,” *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13503–13515, Sep. 2021.

[6] K. Niu, F. Zhang, X. Wang, Q. Lv, H. Luo, and D. Zhang, “Understanding WiFi signal frequency features for position-independent gesture sensing,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4156–4171, Nov. 2022.

[7] R. Gao et al., “Towards position-independent sensing for gesture recognition with Wi-Fi,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–28, 2021.

[8] R. Gao et al., “Towards robust gesture recognition by characterizing the sensing quality of WiFi signals,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–26, 2022.

[9] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, “WiFinger: Talk to your smart devices with finger-grained gesture,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 250–261.

[10] J. Zhang, Y. Li, H. Xiong, D. Dou, C. Miao, and D. Zhang, “HandGest: Hierarchical sensing for robust in-the-air handwriting recognition with commodity WiFi devices,” *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19529–19544, Oct. 2022.

[11] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using WiFi signals,” in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 90–102.

[12] W. He, K. Wu, Y. Zou, and Z. Ming, “Wig: WiFi-based gesture recognition system,” in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2015, pp. 1–7.

[13] R. H. Venkatnarayan, G. Page, and M. Shahzad, “Multi-user gesture recognition using WiFi,” in *Proc. Annu. Int. Conf. Mobile Syst., Appl., Services*, 2018, pp. 401–413.

[14] J. Yang, H. Zou, Y. Zhou, and L. Xie, “Learning gestures from WiFi: A siamese recurrent convolutional architecture,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10763–10772, Dec. 2019.

[15] N. Yu, W. Wang, A. X. Liu, and L. Kong, “QGesture: Quantifying gesture distance and direction with WiFi signals,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–23, 2018.

[16] Y. Zheng et al., “Zero-effort cross-domain gesture recognition with Wi-Fi,” in *Proc. Annu. Int. Conf. Mobile Syst., Appl., Services*, 2019, pp. 313–325.

[17] S. D. Regani, B. Wang, Y. Hu, and K. R. Liu, “GWrite: Enabling through-the-wall gesture writing recognition using WiFi,” *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5977–5991, Apr. 2023.

[18] X. Wang, K. Sun, T. Zhao, W. Wang, and Q. Gu, “Dynamic speed warping: Similarity-based one-shot learning for device-free gesture signals,” in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2020, pp. 556–565.

[19] H. Abdelnasser, M. Youssef, and K. A. Harras, “WiGest: A ubiquitous WiFi-based gesture recognition system,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 1472–1480.

[20] S. Tan and J. Yang, “WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition,” in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 201–210.

[21] X. Li et al., “Crossgr: Accurate and low-cost cross-target gesture recognition using Wi-Fi,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–23, 2021.

[22] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, “Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM,” *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 1011–1021, Apr. 2019.

[23] Y. Song, D. Demirdjian, and R. Davis, “Continuous body and hand gesture recognition for natural human-computer interaction,” *ACM Trans. Interact. Intell. Syst.*, vol. 2, no. 1, pp. 1–28, 2012.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.

[24] P. Wang, W. Li, S. Liu, Y. Zhang, Z. Gao, and P. Ogunbona, “Large-scale continuous gesture recognition using convolutional neural networks,” in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2016, pp. 13–18.

[25] B. Fang, J. Co, and M. Zhang, “DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation,” in *Proc. ACM Conf. Embedded Netw. Sens. Syst.*, Delft, The Netherlands, 2017, pp. 1–13.

[26] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, “A hidden markov model-based continuous gesture recognition system for hand motion trajectory,” in *Proc. Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[27] C. Jian, J. Li, and M. Zhang, “LSTM-based dynamic probability continuous hand gesture trajectory recognition,” *IET Image Process.*, vol. 13, no. 12, pp. 2314–2320, 2019.

[28] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, “A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors,” *IEEE Sensors J.*, vol. 16, no. 16, pp. 6425–6432, Aug. 2016.

[29] W. Chen, L. Chen, M. Ma, F. S. Parizi, S. Patel, and J. Stankovic, “ViFin: Harness passive vibration to continuous micro finger writing with a commodity smartwatch,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–25, 2021.

[30] K. Niu, X. Wang, F. Zhang, R. Zheng, Z. Yao, and D. Zhang, “Rethinking doppler effect for accurate velocity estimation with commodity WiFi devices,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2164–2178, Jul. 2022.

[31] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, “Understanding and modeling of WiFi signal based human activity recognition,” in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 65–76.

[32] D. Zhang, H. Wang, and D. Wu, “Toward centimeter-scale human activity sensing with Wi-Fi signals,” *Computer*, vol. 50, no. 1, pp. 48–57, Jan. 2017.

[33] Y. Zeng, D. Wu, J. Xiong, E. Yi, R. Gao, and D. Zhang, “Farsense: Pushing the range limit of WiFi-based respiration sensing with CSI ratio of two antennas,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–26, 2019.

[34] D. Zhang et al., “Practical issues and challenges in CSI-based integrated sensing and communication,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2022, pp. 836–841.

[35] F. Zhang et al., “Towards a diffraction-based sensing approach on human activity recognition,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 1, pp. 1–25, 2019.

Ruiyang Gao received the B.S. degree from Shandong University, Jinan, China, in 2016, and the Ph.D. degree in computer software and theory from the School of Computer Science, Peking University, Beijing, China, in 2023.

His research interests include wireless sensing, deep learning, and mobile computing.

Wenwei Li received the bachelor’s degree in computer science and technology from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, in 2022, where he is currently pursuing the Ph.D. degree in computer software and theory with the School of Computer Science.

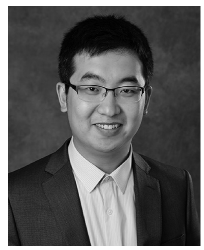
His current research interests include ubiquitous computing, mobile computing, wireless sensing, and deep learning.

Jinyi Liu received the M.E. degree in computer applied technology from the School of Computer Science and Technology, University of Chinese Academy of Science, Beijing, China, in 2016. She is currently pursuing the Ph.D. degree in computer science with the School of Electronics Engineering and Computer Science, Peking University, Beijing.

Her research interests include ubiquitous computing, wireless sensing, and mobile computing.

Shuyu Dai received the B.S. degree from the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, in 2023, where she is currently pursing the Ph.D. degree with the School of Computer Science.

Her research interests include computer vision and wireless sensing.



Mi Zhang (Senior Member, IEEE) received the B.S. degree from Peking University, Beijing, China, in 2006, and the master’s and Ph.D. degrees from the University of Southern California, Los Angeles, CA, USA, in 2010 and 2013, respectively.

He was a Postdoctoral Associate with Cornell University, Ithaca, NY, USA. He is an Associate Professor with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA. His research lies at the intersection of mobile/edge/IoT systems and AI, spanning areas, including on-device AI, federated learning, AI for wireless systems, and mobile health.

Dr. Zhang has received a number of awards for his research. He was the 4th Place Winner of the 2019 Google MicroNet Challenge, the Third Place Winner of the 2017 NSF Hearables Challenge, and the Champion of the 2016 NIH Pill Image Recognition Challenge. He is the recipient of seven best paper awards and nominations. He is also the recipient of the National Science Foundation CRII Award, the Facebook Faculty Research Award, and the Amazon Machine Learning Research Award.

Leye Wang (Member, IEEE) received the Ph.D. degree in computer science from TELECOM SudParis, Évry, France, and University Paris 6, Paris, France, in 2016.

He was a Postdoctoral Researcher with The Hong Kong University of Science and Technology, Hong Kong. He is currently an Assistant Professor with the Key Laboratory of High Confidence Software Technologies (Peking University, China), MOE, and the School of Computer Science, Peking University. His research interests include ubiquitous computing, mobile crowdsensing, and urban computing.

Daqing Zhang (Fellow, IEEE) received the Ph.D. degree from the University of Rome “La Sapienza,” Rome, Italy, in 1996.

He is currently a Chair Professor with the School of Computer Science, Peking University, Beijing, China, and Telecom SudParis, Évry, France. He has published over 300 technical papers in leading conferences and journals. His current research interests include context-aware computing, urban computing, mobile computing, big data analytics, and pervasive elderly care.

Prof. Zhang was a recipient of the Ten-Years CoMoRea Impact Paper Award at IEEE PerCom 2013, the Honorable Mention Award at ACM UbiComp 2015 and 2016, the Ten-Years Most Influential Paper Award at IEEE UIC 2019, and the Distinguished Paper Award at ACM UbiComp 2021. He served as the general or the program chair for over 17 international conferences, giving keynote talks at more than 20 international conferences. He is an Associate Editor of IEEE PERVASIVE COMPUTING, *ACM Transactions on Intelligent Systems and Technology*, and *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*.

Authorized licensed use limited to: Peking University. Downloaded on April 25,2025 at 08:53:32 UTC from IEEE Xplore. Restrictions apply.