# Towards Position-Independent Sensing for Gesture Recognition with Wi-Fi

RUIYANG GAO, Peking University, China
MI ZHANG, Michigan State University, USA
JIE ZHANG, YANG LI, ENZE YI, DAN WU, and LEYE WANG, Peking University, China
DAQING ZHANG*, Peking University, China and Institut Polytechnique de Paris, France

Past decades have witnessed the extension of the Wi-Fi signals as a useful tool sensing human activities. One common assumption behind it is that there is a one-to-one mapping between human activities and Wi-Fi received signal patterns. However, this assumption does not hold when the user conducts activities in different locations and orientations. Actually, the received signal patterns of the same activity would become inconsistent when the relative location and orientation of the user with respect to transceivers change, leading to unstable sensing performance. This problem is known as the position-dependent problem, hindering the actual deployment of Wi-Fi-based sensing applications. In this paper, to tackle this fundamental problem, we develop a new position-independent sensing strategy and use gesture recognition as an application example to demonstrate its effectiveness. The key idea is to shift our observation from the traditional transceiver view to the hand-oriented view, and extract features that are irrespective of position-specific factors. Following the strategy, we design a position-independent feature, denoted as Motion Navigation Primitive(MNP). MNP captures the pattern of moving direction changes of the hand, which shares consistent patterns when the user performs the same gesture with different position-specific factors. By analyzing the pattern of MNP, we convert gestures into sequences of strokes(e.g, line, arc and corner) which makes them easy to be recognized. We build a prototype WiFi gesture recognition system,i.e., *WiGesture* to validate the effectiveness of the proposed strategy. Experiments show that our system can outperform the start-of-arts significantly in different settings. Given its novelty and superiority, we believe the proposed method symbolizes a major step towards gesture recognition and would inspire other solutions to position-independent activity recognition in the future.

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

Additional Key Words and Phrases: Gesture Recognition, Wireless Sensing, Channel State Information (CSI)

**ACM Reference Format:**
Ruiyang Gao, Mi Zhang, Jie Zhang, Yang Li, Enze Yi, Dan Wu, Leye Wang, and Daqing Zhang. 2021. Towards Position-Independent Sensing for Gesture Recognition with Wi-Fi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 61 (June 2021), 28 pages. https://doi.org/10.1145/3463504

*This is the corresponding author

Authors' addresses: Ruiyang Gao, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, gry@pku.edu.cn; Mi Zhang, Michigan State University, Michigan, USA, mizhang@msu.edu; Jie Zhang; Yang Li; Enze Yi; Dan Wu; Leye Wang, School of Electronics Engineering and Computer Science, Peking University, Beijing, China; Daqing Zhang, Key Laboratory of High Confidence Software Technologies (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing, China , Telecom SudParis, Institut Polytechnique de Paris, Evry, France, dqzhang@sei.pku.edu.cn.


2474-9567/2021/6-ART61 $15.00
https://doi.org/10.1145/3463504

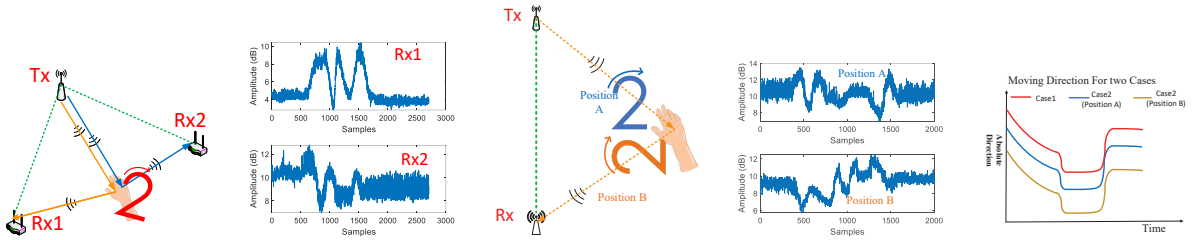
## 1 INTRODUCTION

With the rapid development of wireless technologies, the growing capabilities of Wi-Fi have made it possible to utilize Channel State Information (CSI) for sensing a wide range of human activities. Given that there is no need for hardware modification, Wi-Fi-based sensing technologies have been developed with various applications including elderly monitoring [24, 37, 40], daily activities recognition [15, 18, 26, 29, 30] and gesture recognition [1, 20, 21, 41]. Despite its popularity, existing Wi-Fi-based sensing technologies are constrained by a practical challenge which we refer to as the *position-dependency* challenge: many existing solutions use machine learning or deep learning-based techniques to learn and extract features from the CSI signals that can be used to recognize human actions and activities. Unfortunately, these features capture information related to not only the action itself but also the location and orientation of the action [12, 32, 39]. Therefore, existing solutions can not recognize actions in a location/orientation-independent manner. As a result, they often require intensive training with data collected at different locations and orientation, but may undergo performance drop at new locations and orientations that are not included in the training dataset.

Recently, Wi-Fi-based gesture recognition emerges as a very promising Wi-Fi-based sensing technology. Many existing works mainly use machine learning and deep learning models which learn CSI waveform features for recognizing hand gestures. For example, Mudra [41] recognizes gestures by comparing features in frequency distribution for different gestures. WiFinger [20] identifies different gestures by extracting features in the time domain and compares waveforms using Dynamic Time Warping (DTW). WiMU [21] leverages frequency features extracted by short-time Fourier transform to generate virtual samples to enable multi-user gesture recognition. WiGest [1] recognizes gestures using patterns of changes in primitives, such as rising edges, falling edges, and pauses. Unfortunately, these systems all leverage features extracted in a specific setting with fixed location and orientation where location and orientation are defined as the relative location and input angle of the hand related to the Wi-Fi transceivers. Given that, these systems are constrained by the position-dependency challenge.

To address this position-dependency challenge, a number of approaches have been developed recently. In particular, WiAG [22] analyzes how the hand moves relative to the transceivers to develop a translation function to generate virtual samples for gestures in all the possible locations and orientations. WiDar3.0 [44] uses velocity profiles of gestures extracted from Doppler Frequency Shift (DFS) which acts as unique indicators of gestures for recognition. Although these works have made great progress, they all exploit the measurements which represent how the hand moves relative to the transceivers (we denote it as the “transceiver view”), which are still inconsistent when transceivers are deployed at different locations. This can be observed in Figure 1. As shown in Figure 1(a) and Figure 1(c), an user is drawing digit ’2’ at different locations/orientations and two receivers (Rx1, Rx2) are deployed at two different locations to record CSI signals. Given that the location and the orientation of the same gesture captured at the two receivers are different, the CSI waveform (Figure 1(b) and Figure 1(d)) recorded on the two receivers are clearly inconsistent. Therefore, to enable position-independent sensing, these works [22, 44] need to acquire extra position-specific information including the device locations, initial locations and orientations of the hand. Such requirements are usually impractical in real-world use cases.

To address the limitations of existing works, in this paper, we propose a new position-independent approach and use gesture recognition as an application example to demonstrate its effectiveness. The core of our approach is that instead of depicting hand motions from the transceiver view, a gesture can be consistently characterized by how the hand moves relative to its previous position (which we denote as a “hand-oriented view"). Such a hand-oriented view can provide consistent and distinguishable representations for different hand motions that are irrelevant to the location and the orientation of the hand with respect to the device deployment. For the example in Figure 1(a) and Figure 1(c), no matter how we deploy our transceivers, the hand movement can be consistently described from the hand-oriented view as four segments: {*go around clockwise, go straight, turn counterclockwise and go straight*}.




(a) Case 1:Drawing '2' with unfix-deployed transceivers

(b) CSI waveforms for Case 1

(c) Case 2:Drawing '2' with different locations and orientations

(d) CSI waveforms for Case 2

(e) Moving direction changes for two cases

Fig. 1. Illustrations of Drawing '2' with Different Position-specific Factors

Following this idea, we extract features that contain hand-oriented view-based information. The moving direction of the hand from the hand-oriented view, could be depicted in Figure 1(e), the moving direction change is found to be consistent for the same gesture, no matter where the gesture is performed. Inspired by this insight, we exploit the moving direction of the hand, and incorporate a unique primitive feature which we refer to as *Motion Navigation Primitive* (MNP). With MNP, we are able to identify the in-air gestures with users moving their hand in a predefined trajectory, without the need to know the exact locations of the hand and Wi-Fi transceivers.

In summary, our paper makes the following contributions.

- To tackle the *position-dependency* challenge, we propose a new position-independent approach that extracts features from the hand-oriented view instead of the traditional transceiver-oriented view. Our proposed approach provides consistent position-independent knowledge without the need for intensive training under different locations and orientations.
- Based on the hand-oriented view, we design a concrete position-independent feature, denoted as Motion Navigation Primitive (MNP), which captures the pattern of moving direction changes of the gesture. By analyzing the pattern of MNP, we can convert a set of gestures into a sequence of strokes that make the gestures easy to be recognized.
- To validate the effectiveness of MNP and the hand-orient view-based strategy, we build a prototype system named *WiGesture* to recognize various gestures under diverse environments with different device locations, users, room layout, etc. Our experiments show that our system with MNP outperforms status quo systems when users perform gestures with different position-specific factors.

The remaining of the paper is organized as follows. We provide the background in Sec 2. In Sec 3, we introduce the hand-oriented view-based sensing strategy to alleviate the position-dependency challenge and explore a position-independent feature (MNP) for gesture recognition. We describe how to extract the MNP in Sec 4, and how to recognize gestures using MNP in Sec 5. In Sec 6, we describe the prototype design and conduct various experiments to evaluate the effectiveness of our proposed approach. Sec 7 discusses the limitation and opportunities. Sec 8 surveys the related works. Finally, we conclude our work in Sec 9.

## 2 BACKGROUND

In this section, we provide a brief background on CSI and the Fresnel zone model.

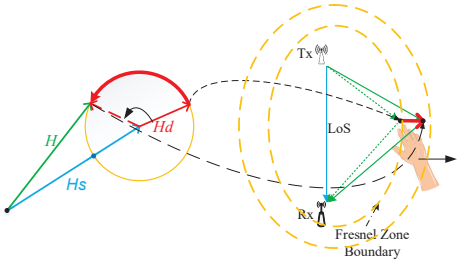



Fig. 2. Illustration of CSI dynamics using Fresnel Zone Theory.

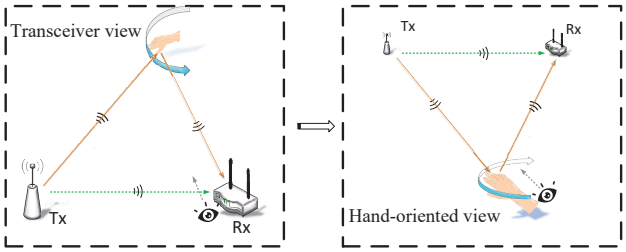



Fig. 3. Shift observation from transceiver view to hand-oriented view.

## 2.1 Sensing Hand Motions with CSI

State-of-the-art Wi-Fi-based gesture recognition systems extract CSI signals from wireless cards to sense gesture motions. When the gesture is being performed, the CSI signal will propagate among multiple paths in the environment as the transmitter sends continuous Wi-Fi signals [24, 30, 33]. The received CSI signal is the superposition of all the path components, including the static Line-of-Sight (LoS) propagation, the environment reflections, and the dynamic reflection path of the hand. The received CSI signal can thus be decomposed into static and dynamic components where the static component is affected by the surrounding environment and the LoS of transceivers while the dynamic component is determined by the reflection path from the hand. We can denote the CSI as following equation:

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

where $H_s(f,t)$ is the static phasor component; $\lambda$ is the wavelength; $A(f,t)$, $e^{-j2\pi\frac{d(t)}{\lambda}}$, and $d(t)$ are the attenuation, the phase shift, and the path length of the dynamic phasor component $H_d(f,t)$, respectively.

## 2.2 Fresnel Zone Model

Fresnel zone model [8, 24, 25, 34, 39] divides the space into a series of concentric ellipsoids of alternating strengths in a pair of transceivers. They are caused by a radio wave following multiple paths as it propagates in space. As the hand moves, the reflection path change simultaneously. For every Fresnel Zone layer penetrated by the hand, the reflection path change $\lambda$, and the phase of the dynamic component changes $2\pi$. When the dynamic component travels along the reflection path with the length of $d$, its phase is shifted by $2\pi d/\lambda$ as shown in Figure 2. We refer to the phase of the dynamic component as the dynamic phase.

While this theory models how signals propagate in the free space and make impacts on received CSI, it also reveals an important phenomenon: CSI waveform patterns can be different for similar hand motions under different position-specific factors. It depends on how many Fresnel Zones the hand moves across when performing the gesture as well as the relative location and orientation to the transceivers in the Fresnel Zone [39].

# 3 SENSING GESTURE WITH HAND-ORIENT VIEW BASED STRATEGY

In this section, we first illustrate the fundamental position-dependency problem which exists in existing gesture recognition systems. And we introduce how to address this critical limitation by hand-oriented view-based strategy. Then we give the intuition of the position-independent feature, i.e., MNP which is built upon the hand-oriented view. Finally, we briefly describe how to extract this feature.

## 3.1 Hand-oriented View-based Strategy

*3.1.1 Position-dependency Problem in Gesture Recognition.* The *Position-dependency* problem is a common issue of existing gesture recognition systems. The wireless signals are not only determined by hand movement but also highly related to position-specific factors [12, 32, 39].

Early gesture recognition solutions [1, 20, 21, 41] leverage statistical patterns (e.g., waveform, frequency distribution) from CSI to map gestures with signals. However, as we mentioned in Sec.2.2, depending on the relative location and orientation of the hand to the transceivers, CSI waveform patterns can be different for similar hand motions. Therefore, pattern-based approaches essentially rely on intensive training to achieve good results. Since it is difficult to collect the CSI waveforms with all gesture locations and orientation for training, their performance cannot be guaranteed.

Another set of solutions [22, 44] explore physical features (e.g., DFS, AoA, ToF) behind CSI and generate unique profiles of gestures. Unfortunately, these features depict how the hand moves relative to the transceivers. They are still inconsistent when the position of the hand or the device changes. Therefore, without the help of extra information such as the device deployment locations and the initial location of the hand, these solutions could fail to recognize the same gesture with different locations and orientations.

*3.1.2 Relieving Position-dependency Problem from Hand-oriented View.* The drawbacks of existing works ask for a new strategy to extract a position-independent profile of the gesture without requiring any extra position-specific information. To this end, we shift our view from transceivers to hand itself. As shown in Figure 3, when hand movements are observed from the transceivers (i.e., transceiver view), the corresponding observation is a description of the relative positional relationship between the hand and the device, and is inconsistent with changing location and orientation of the hand as well as the location of devices. However, when observing from the hand perspective, the gesture can be characterized as a sequence of physical descriptions about how the hand moves relative to its previous position. From this hand-oriented view, the observation is consistent no matter how users perform gestures in different locations and input angles. It is also irrelevant to the location of the devices.

As received signals (i.e., CSI) is transceiver view based, it is still challenging to shift our observation to a hand-oriented view. Since one pair of transceivers is inadequate to capture the position-independent profile of the gesture, for a gesture with a 2D trajectory, we need two observations from different transceiver's views to complementary depict how the hand moves. The kinetic representation of the gesture in the hand-oriented view can be derived from multiple observations from the transceiver view. We will dive deeper by exploring a unique position-independent feature below.

## 3.2 MNP: A Position-Independent Feature from Hand-orient Views

*3.2.1 Intuition of MNP.* We now consider which specific feature can be derived and is capable of position-independent gesture recognition. Intuitively, when observing the hand movement from a hand-oriented view, we note that the patterns on how the moving direction of the hand changes relative to its previous position is consistent for the same gesture with different locations and orientations (shown in Figure 1(e)). As such, we extract a unique position-independent feature, i.e., Motion Navigation Primitive, which indicates how the moving direction of the hand changes to identify in-air gestures with predefined trajectories. Gestures with distinct moving direction changes will have unique patterns in MNP, regardless of their location and orientation.

*3.2.2 Derivation of MNP.* It is difficult to directly calculate the moving direction changes because the velocity of the hand at each moment is unknown. However,the velocity of the gesture can be projected into two velocity components along with any two non-parallel directions. When moving direction changes, we find that the ratio of these two velocity components will change synchronously. If we can estimate the ratio of velocity components, we can depict how the moving direction of the hand changes.

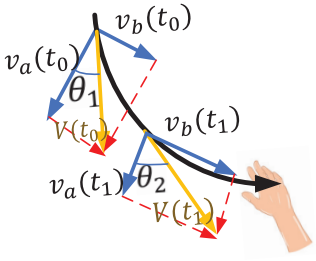



Fig. 4. The moving direction can be indicated by ratio of two velocity components.

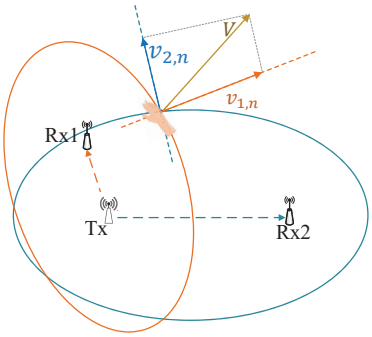



Fig. 5. Projecting hand's velocity along normal directions of ellipses on two links.

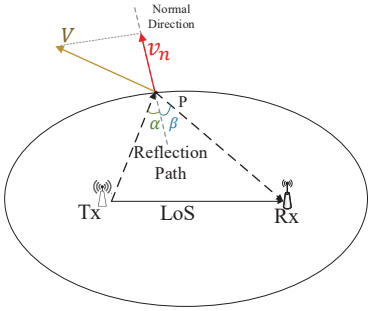



Fig. 6. Relationship of the reflection path and the velocity component along the normal direction .

Figure 4 provides an intuitive analysis. The direction of velocity of the hand changes from $V(t_0)$ at time $t_0$ to $V(t_1)$ at time $t_1$. It is clear that its projections in two fixed directions change accordingly from $v_a(t_0)$, $v_b(t_0)$ to $v_a(t_1)$, $v_b(t_1)$, which also results in the changes of their ratio. When the direction of $v$ changes towards counterclockwise, the value of $\frac{|v_b|}{|v_a|}$ increases at same time. Similarly, the value of $\frac{|v_b|}{|v_a|}$ decreases as the direction of $v$ changes clockwise. The faster the direction of $V$ changes, the faster the ratio changes. Note the V can project into any two non-parallel directions, but during gesture performing the directions should keep unchanged. In the appendix A.1 we give the mathematical justification to elaborate this phenomenon.

In our implementation, we deployed two links (e.g., Tx-Rx1, Tx-Rx2 in Figure 5). Note that the links can be deployed freely as long as each link is not parallel to another. For each link, we construct an ellipse with foci transmitter and receiver as shown in Figure 5. Previous work [13] shows that Wi-Fi signal are mainly effected by the velocity component along the normal directions of the ellipse. Therefore, we project the hand's velocity $V$ into the normal directions of the two ellipses, which are denoted as $v_{1,n}$ and $v_{2,n}$. As the moving range of the hand is far less than the LoS, we assume directions of $v_{1,n}$ and $v_{2,n}$ keep unchanged. Thus their ratio can depict the moving direction changes of the hand.

But without knowing the exact locations of the hand and transceivers, we cannot directly obtain the projected velocity components [9, 17]. To simplify this process, we use the reflection path changes to replace the projected $v_n$ on each link. As shown in Figure 6, only the $v_n$ leads to the reflection path changes $\Delta d$ in $\Delta t$ [13].The length changes for the line $|Tx\text{-}P|$ is $v_n cos\alpha \Delta t$, and the length changes for the line $|Rx\text{-}P|$ is $v_n cos\beta \Delta t$. As the path length change speed is the sum of the speeds in lines $|Tx\text{-}P|$ and $|Rx\text{-}P|$, the relationship between $\Delta d$ and $v_n$ is represented as follows:

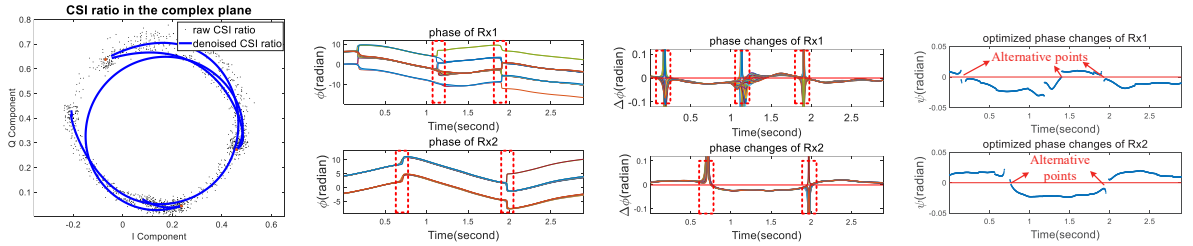
$$|\frac{\Delta d}{\Delta t}| = |v_n|(\cos\alpha + \cos\beta) \tag{2}$$

where $\alpha$ and $\beta$ is the angles between $v_n$ and the reflection path shown in Figure 6. As the moving range of the hand is far less than the length of the LoS, $\cos\alpha + \cos\beta$ can be regarded unchanged. As such, at time $t$, we have the following equation:

$$\frac{\Delta d_2(t)}{\Delta d_1(t)} \propto \frac{v_{2,n}(t)}{v_{1,n}(t)} \tag{3}$$

where $d_1(t)$ and $d_2(t)$ are the length of reflection path from two pairs of deployed transceivers.

Therefore, the ratio of reflection path changes of two links can replace the ratio of the two velocity components to depict the moving direction changes. We define it as the MNP. Specifically, MNP $M(t)$ in the time $t$ is defined




(a) raw and de-noised CSI ratio of one receiver (b) dynamic phase of two receivers (c) dynamic phase changes of all sub-carriers (d) optimized phase changes of all sub-carriers

Fig. 7. Extract Dynamic Phase Changes From CSI ratio

as the following equation:

$$M(t) = \arctan \frac{\Delta d_2(t)}{\Delta d_1(t)} \tag{4}$$

To avoid outliers when the denominator is very small, we applied an *arctan* function on the ratio. Now we analyze how to derive the MNP and connect it with the CSI. Based on the Equ. 1, the length of the reflection path $d(t)$ can be represented as follows:

$$d(t) = \frac{\lambda}{2\pi} \angle \left( \frac{H(f,t) - H_s(f,t)}{A(f,t)} \right) \tag{5}$$

As the moving range of the hand is very small to the LoS, we assume that the $H_s(f,t)$ and $A(f,t)$ keep unchanged during the gesture performing. Therefore, $\Delta d(t)$ can be represented by a function of CSI $H(f,t)$ as follows.

$$\Delta d(t) = \frac{\lambda}{2\pi} \angle \Delta H(f,t) \tag{6}$$

With Equ. 4, MNP can be derived from the CSI as follows:

$$M(t) = \arctan \frac{\angle \Delta H_2(f,t)}{\angle \Delta H_1(f,t)} \tag{7}$$

Where the $H_1(f,t)$ and $H_2(f,t)$ are the received CSI on two pairs of transceivers. According to our assumption, CSI is only affected by its dynamic phasor component. Thus the $\angle \Delta H_1(f,t)$, $\angle \Delta H_2(f,t)$ are equal to the phase changes of the CSI's dynamic phasor components (i.e., dynamic phase changes) on two receivers. Therefore, by exacting the dynamic phase changes from received CSI, MNP can be acquired with two pairs of transceivers in any non-parallel displacement without knowing their locations in advance.

## 4 MNP EXTRACTION

This section provides details about how to extract MNP from CSI data. Technically we take three steps to achieve this goal. Firstly we exploit the dynamic phase changes of the CSI. Secondly, MNP is estimated by the ratio of dynamic phase changes on two pairs of transceivers. Finally, we enhance the MNP by removing the segments affected by handshaking.

### 4.1 Extract the Dynamic Phase Changes from the CSI

It faces many problems to obtain the dynamic phase changes of CSI. The raw CSI signal collected on the receiver contains random phase offset including Channel Frequency Offset (CFO) and Sample Frequency Offset (SFO),

which makes it hard to extract CSI phase information directly [35]. To address this problem, we leverage the CSI ratio [12, 32, 38]. The CSI ratio has the form in the following equation:

$$H_q(f,t) = \frac{A_{noise}(f,t)e^{-j\theta_{offset}(f,t)}(H_{s1}(f,t) + H_{d1}(f,t))}{A_{noise}(f,t)e^{-j\theta_{offset}(f,t)}(H_{s2}(f,t) + H_{d2}(f,t))} = \frac{H_{s1}(f,t) + H_{d1}(f,t)}{H_{s2}(f,t) + H_{d2}(f,t)} \tag{8}$$

where $A_{noise}$ is the impulse noise in CSI amplitude, $\theta_{offset}$ is the random phase offset such as Channel Frequency Offset (CFO) and Sample Frequency Offset (SFO). $H_{s(1,2)}(f,t)$ are the static components and $H_{d(1,2)}(f,t)$ are the dynamic components for the two antennas of the same receiver. It has been proved in [32, 38] that using the CSI ratio can suppress the impulse noise in amplitude, eliminate these phase random offsets and approximately recover CSI in the complex plane. With tolerable inaccuracy, the dynamic phase of CSI can be roughly expressed by the rotation angle $\varphi$ of the CSI ratio. Every time the rotation angle changes $2\pi$, the dynamic phase will also change $2\pi$ . In the appendix A.2 we further elaborate how CSI raito captures the hand movements through an experiment.

We still face two challenges to extract the dynamic phase using the CSI ratio. First, the origin samples on the complex plane are dispersed and the rotation angles of the CSI ratio are hard to extract. To address this challenge, we adopt Savitzky-Golay(S-G) [34] filter to smooth the samples which form the circular shape of CSI ratio. Figure 7(a) shows the origin and denoised CSI ratio samples of the gesture of digit '2'. Second, as the circular center of the arcs in the CSI ratio is continuously changing, it is difficult to determine the rotation angle and calculate the dynamic phase $\varphi$ accurately. To address this challenge, we measure the slope change of the tangent line of the de-noised CSI ratio on the complex plane to calculate $\varphi$. This is effective as it only involves the subtraction of the two consecutive data of the CSI ratio.

Now on we can obtain the dynamic phase changes $\Delta\varphi$ by getting the differential of the dynamic phase. However, when the direction of reflection path changes alters (e.g., the length of reflection path firstly increases then decreases), the clock directionality of the CSI ratio rotation will also change. And a phase jump of $\pi$ is introduced in the dynamic phase by calculating the tangent slope. Figure 7(b) shows the extracted dynamic phase $\varphi$ of all sub-carriers. The dynamic phase changes are shown in Figure 7(c). Each sub-carrier is coated with a different color. We also mark the positions of the phase jumps in the red box. We can observe obvious outliers caused by phase jumps of red marks. We thus prune these phase jumps by eliminating where phase changes dramatically.

The extracted $\Delta\varphi$ contains data streams of 30 sub-carriers. All the sub-carriers provide similar information but with varying signal quality. Picking information from the right sub-carrier can improve the estimation reliability. To score the sub-carriers and select the signal of the best quality. We apply a dynamic time-slicing selection scheme.

We denote the dynamic phase differential $\Delta\Phi$ as $\Psi$. Specifically, we split it into $n$ slices by time windows $\tau$ of $0.25s$. Thus the $\Psi$ can be expressed as follows:

$$\Psi_{all} = \left\{\Psi_{t_0}, \Psi_{t_0+\tau} \ldots \Psi_{t_0+n\tau}\right\} \tag{9}$$

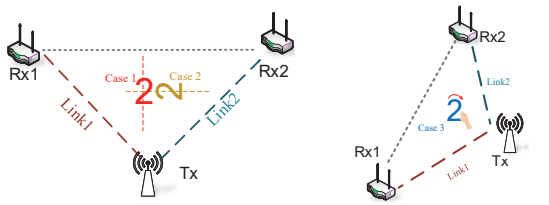
where $t_0$ is the initial time, and $\Psi_t$ denotes a piece of dynamic phase changes starting at time $t$ and lasting $\tau$.

$\Psi_t$ contains information from 30 sub-carriers. We choose the sub-carrier with the lowest variance. Thus we can calculate each $\Psi_t$ as follows:

$$\Psi_t = argmin \frac{var(\psi_{i,t})}{\sum_{i=1}^{30} var(\psi_{i,t})} \tag{10}$$

where $\psi_{i,t}$ denotes the slice of dynamic phase changes on $i$-th sub-carriers at time $t$.

By assigning better scores for smoother phase differential variations, we can select the best signal, i.e. the best sub-carrier piece-by-piece from all candidate sub-carriers, and finally extract the dynamic phase changes with high quality. Figure 7(d) shows the extracted dynamic phase changes.




(a) Drawing '2' with different locations and orietations.

(b) Drawing '2' with another deployment of transceivers

Fig. 8. Drawing '2' in Different Scenarios

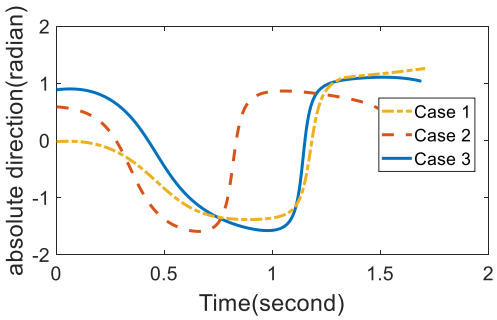



Fig. 9. MNP for Three Cases of Drawing '2'.

## 4.2 Estimate the MNP from the Dynamic Phase Changes

With Equ. 7, now MNP series $M$ can be extracted by calculating the ratio of the dynamic phase from two receivers as shown as follows:

$$M = \arctan \frac{\psi_2}{\psi_1} \tag{11}$$

where $\psi_1$,$\psi_2$ are dynamic phase changes of two receivers respectively.

However, the above-mentioned MNP extraction method takes effect based on the assumption that users move their hand smoothly along the predefined trajectory of the gesture. However when put into practice, the hand and arm may shake at sometime, which can result in the moving direction changes irrelevant to the gesture. It is impractical to avoid any shaking during hand movement. Therefore we remove these affected segments to enhance the robustness of MNP.

To achieve this, we notice that when user shakes his hand, the moving direction goes back and forth during a short time interval, which shows several continuous non-monotone short segments in MNP series. To eliminate these shaking-influenced segments, we leverage the corresponding dynamic phase to analyze the monotonicity of the MNP series. Specifically,as shown in Figure 7(d), we can mark where the symbols of $\Delta\Phi$ change on at least one receiver, and express them as the alternative points. Between these points, the dynamic phase $\Phi$ only increase or decrease. The MNP $M$ can be divided by these alternative points into $\{m_1, m_2 ... m_n\}$. As the original MNP is calculated by the ratio of $\Delta\Phi$ from two receivers, each corresponding $M$ is also monotonous. If hand shakes, the direction of dynamic phase changes will change at least twice, which introduces two consecutive alternative points in a short time. Therefore, the $m$ affected by the hand shake usually appears as a very short segment, which have the opposite monotonicity compared with adjacent segments and are easy to find.

After removing these affected segments, we obtain the MNP series $M$ from the remaining $m$ segments. We adjust the value of the remaining segments to eliminate gaps introduced from deleted segments. After that, we apply the S-G filter on the whole $M$ to make it smoother. An optimized MNP example is shown in Figure 21. Through these measures, we greatly enhance the robustness of MNP. Although some information from the original MNP is discarded, the rest is sufficient to recognize gestures.

# 5 GESTURE RECOGNITION MECHANISM WITH MNP

In this section, we present how to recognize gestures using MNP. First, we demonstrate the feasibility of the MNP through specific examples. Then we analyze how to use the MNP series to recognize different motions (i.e., strokes) of hand movements, followed by the empirical studies to verify the analysis. After that, we discuss how to recognize gestures by decomposing them into a temporal sequence of basic strokes.

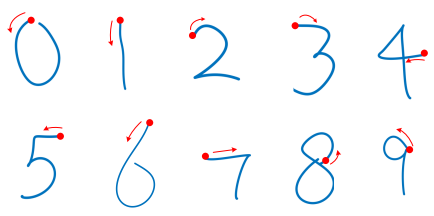


Fig. 10. Illustration of ten digits gestures. Red dots mark where to start drawing.

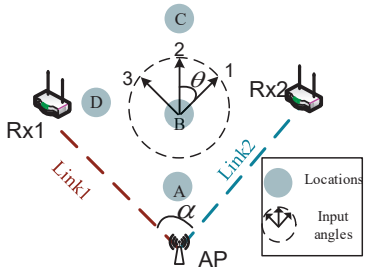



Fig. 11. Illustration of the empirical verification

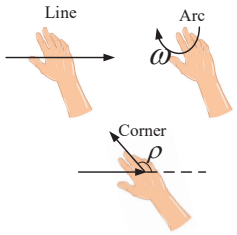



Fig. 12. Illustration of different strokes

## 5.1 Feasibility of Using MNP to Recognizing Gestures

*Repeatability* and *discernibility* are two key requirements for using MNP to recognize gestures. Repeatability means that MNP extracted from the same gesture performed in different scenarios should have similar signal patterns. Discernibility, on the other hand, means that different gestures with different moving direction changes are distinguishable in the patterns of MNP. In this subsection, we show the *repeatability* and *discernibility* of using MNP for gesture recognition.

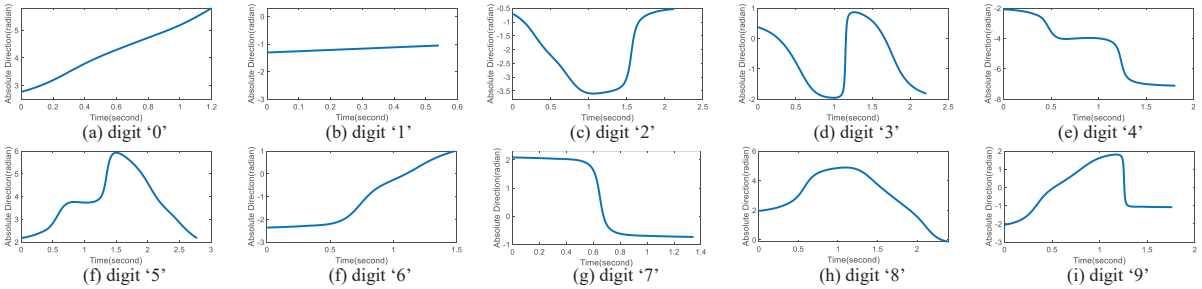
Specifically, for the same gesture performed in different locations and orientations, as MNP characterizes the moving direction changes of the hand, it shows consistent patterns. For example, Figure 8 shows the experiment settings for writing '2' in three different ways. The orientation and location of the gesture are different for Case 1 and 2. And transceivers are deployed differently in Case 3. Figure 9 shows the extracted MNP corresponding to '2' in Figure 8, apparently the MNP patterns are quite consistent. On the other hand, for different gestures in digit sets in Figure 10, as each gesture corresponds to a unique moving direction changing pattern and is different from others, their MNP patterns are unique and different (Figure 13).

Therefore, we analyze the MNP pattern and convert it into a sequence of different action modes (i.e., strokes) to recognize each gesture. This strategy not only applies to the digit sets but also applies to other gesture sets as shown in Sec 6.2. We will elaborate this strategy in the following subsection.

## 5.2 Identify Basic Strokes with MNP

To recognize gestures, one standard way is to learn a classifier using MNP as input features. However, the development of the classier still requires a large amount of training data. To avoid this overhead, we propose a new gesture recognition mechanism.

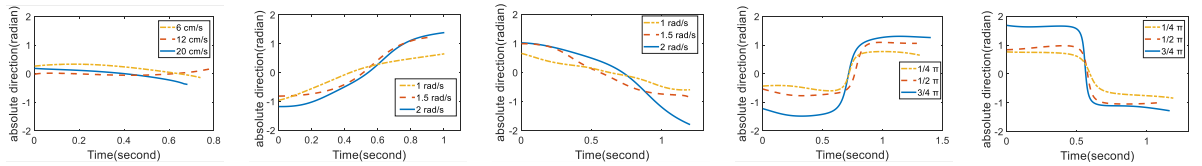
As previously mentioned, gestures can be discriminated by the overall pattern of the MNP. But it is still a problem to characterize these patterns. We note that MNP can be represented by multiple segments with different trends (e.g., rising, failing, or unchanged). As the MNP captures the pattern of moving direction changes, if the moving direction changes fast, the corresponding MNP series also show a trend of rapid change and vice versa. Moreover, whether MNP is increasing or decreasing is determined by whether the direction changes clockwise or counterclockwise. In other words, difference trends in MNP patterns indicate how fast the moving direction changes and its clockwise directionality in the gesture. Therefore, we can further divide these segments into several simple and instinct categories, and the categorized segments can be physically linked to a basic action mode in terms of the moving direction changes. We define these basic action modes as three different strokes, namely, line, arc and corner. And a complicated gesture can be converted into the sequence of these strokes. Apparently, as long as the sequence of strokes for each gesture is different, those gestures can be accurately recognized.




(a) digit '0' (b) digit '1' (c) digit '2' (d) digit '3' (e) digit '4'

(f) digit '5' (f) digit '6' (g) digit '7' (h) digit '8' (i) digit '9'

Fig. 13. Motion Navigation Primitive for 10 digits




(a) type-line stroke (b) type-arc in counter-clockwise (c) type-arc in clockwise (d) type-corner in counter-clockwise (e) type-corner in clock-wise

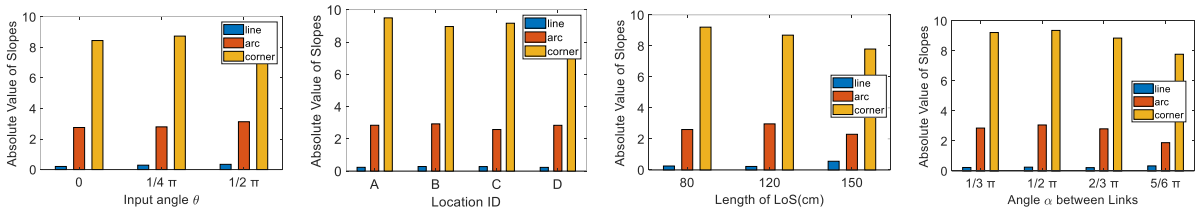
Fig. 14. MNP variances for different basic strokes

Specifically, we use $[t_0, t_1]$ to represent a small time duration when users performing a gesture. And we use $\delta\theta$ to denote the moving direction changing rate during $[t_0, t_1]$. Three different strokes are listed as follows.

(1) **Type-Line**: $\Delta\theta$ is less than a small threshold $k_1$.In that case, we believe that the moving direction of the hand has hardly changed and users are waving a line.
(2) **Type-Arc**: $\Delta\theta$ is larger than the $k_1$ but smaller than a very high threshold $k_2$. We believe that the moving direction of the hand monotonically and moderately changes and users are waving an arc.
(3) **Type-Corner**: $\Delta\theta$ is larger than the threshold $k_2$, which means the moving direction of the hand has undergone a very quick change, and the users are drawing a corner.

Note that $k_1$,$k_2$ are adjustable. They do not need to be an exact and constant value as long as we can use them to decompose gestures and make them distinguishable. According to the ground truth in the empirical experiments in the following subsection, the $k_1$,$k_2$ can be roughly set to 0.3 rad/s and 6 rad/s.

Based on our previous analysis, each type of stroke has a different tendency on the MNP series as shown in Figure 14. Therefore, the slope information of MNP can be used to identify different basic gesture strokes. Similarly, we classify the slope into three categories according to their magnitude. Specifically, the slope with a small magnitude corresponds to a type-line stroke. The slope with a large magnitude corresponds to a type-corner stroke. And others are type-arc strokes. Besides, the clock directionality of strokes can also be extracted from the MNP tendency. When the MNP series shows an increasing trend, the direction of gesture stroke changes counterclockwise. Similarly, when MNP decreases, the direction of gesture stokes changes clockwise. We will verify this method with empirical studies and explore specific classification thresholds for the slopes of MNP in the next subsection.




(a) input angles (b) performing locations (c) LoS (d) angles between links

Fig. 15. Average of Absolute Value of Slope on MNP by different factors

## 5.3 Verification of MNP to Identify Strokes

In this section, we first discuss the variance of MNP when performing a specific type of stroke. Then we conduct experiments to empirically verify the capability of using MNP to identify three types of basic strokes.

*5.3.1 Variances of MNP for Different Strokes.* The user's hand movement is difficult to be exactly the same for a same type of stroke performing at different times. Hence it is necessary to examine how the pattern of MNP varies for different actions. Generally, as the MNP patterns indicate the moving direction changes, faster the moving direction changes, larger the absolute value of the slope of the MNP curve will be.

Figure 12 shows the basic motions of three types of strokes. And Figure 14 demonstrates the MNP series of basic strokes. When the user performs a type-line stroke, the direction rarely changes. The curve of MNP keeps a line with a low and slightly changed slope. As shown in Figure 14(a), the user is waving a line of 6cm/s, 12cm/s and 20cm/s respectively, the pattern of MNP is very similar. When the user performs a type-arc stroke with different radian changing speed $\omega$ shown in Figure 12, the slopes of MNP are different. High radian change speed can result in a large slope. Figure 14(b) and Figure 14(c) show the patterns of MNP with three radian changing speeds $\omega$ around 1 $rad/s$,1.5 $rad/s$ and 2 $rad/s$. When the user performs a type-corner stroke with different angles $\rho$ shown in Figure 12, the slopes of MNP are also different. A sharp corner can lead to a large slope. Figure 14(d) and Figure 14(e) show the patterns of MNP with three different corners with the angle $\rho$ of 1/4 $\pi$, 1/2 $\pi$ and 3/4 $\pi$.

From the above examples, we can see that although the MNP series for a same stroke type are not completely the same, the slopes corresponding to different strokes are quite distinguishable.

*5.3.2 Empirical Verification.* To verify the performance of the MNP in identifying three types of basic strokes through threshold-based strategy under various settings, we conduct the empirical experiments. Specifically, we place one transmitter(Tx) to send WiFi signals and two receivers(Rx) to record CSI as shown in Figure 11. Participants need to perform these three basic gestures as shown in Figure 12. In order to make it more reliable and practical, we do not require participants to move their hand with a precise trajectory. For each data, we generate the corresponding MNP series and calculate the average slopes. We evaluate the effects of the following four factors on MNP recognition ability. For convenience, the orientations of the gesture and the angle between two links are denoted as $\theta$ and $\alpha$ respectively.

*Overall results.* We collect 300 samples totally in each type of the stroke for different factors. Figure 16 shows their statistics of slopes in the corresponding MNP series. We observed that these data grouped into three categories related to their types of strokes.

*Different orientations.* There are three orientations in our experiment. We mark the first orientation as 0 where hand moves approximately parallel to the LoS of link1 at the start. After that the moving angle rotates $\theta$ of $\frac{1}{4}\pi$ and $\frac{1}{2}\pi$ when subject performing strokes with second and third orientations at start. Thus the second and third

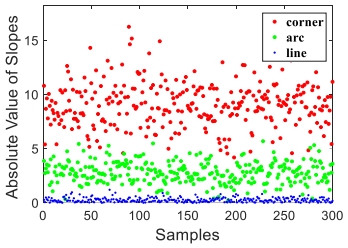



Fig. 16. Overall Distribution of MNP Slopes

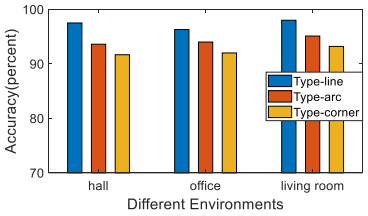



Fig. 17. Accuracy with Selected Thresholds to Identify Strokes

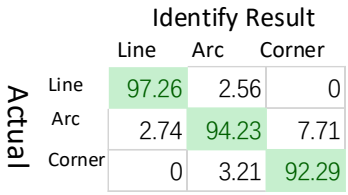



Fig. 18. Confusion Matrix of Selected Thresholds

orientations are marked as $\frac{1}{4}\pi$ and $\frac{1}{2}\pi$ respectively. Each type of strokes is performed 20 times in each setting. During the experiment, other factors remain unchanged. The LoS is 80$cm$. The $\alpha$ is $\frac{1}{2}\pi$ and gestures are performed in area A. Figure 15(a) shows the average of the absolute value of slopes on MNP. The slopes for different strokes are distinguishable with all orientations.

*Different performing locations.* We perform basic strokes in four different locations(A-D) shown in Figure 11. Both $\theta$ and $\alpha$ are $\frac{1}{2}\pi$, and the LoS is 80$cm$. Each type of stroke is performed 20 times in each setting. Figure 15(b) shows the related results. The distinction of strokes' slope is stable in different areas.

*Different Line-of-Sight(LoS).* We set three different distances of LoS which are 80$cm$, 120$cm$ and 120$cm$. Both $\theta$ and $\alpha$ are $\frac{1}{2}\pi$ and users are performing strokes in area A. Each type of strokes is performed 20 times in each setting. Figure 15(c) shows the results. We can find that the value of the slope is slightly descending with longer LoS. However, it is still easy to distinguish using a threshold.

*Different angles of two links.* We consider four different Different angles $\alpha$ of two links. The $\alpha$ is set for $\frac{1}{3}\pi$, $\frac{1}{2}\pi$, $\frac{2}{3}\pi$ and $\frac{5}{6}\pi$ respectively. The $\theta$ is $\frac{1}{2}\pi$ with LoS of 80$cm$. Users are performing strokes in area A. Similarly, each type of strokes are performed 20 times in each setting. The results are shown in Figure 15(d). Still the types of strokes are easily to determined by slopes of MNP. From the results we can conclude that, in most situations, the generated MNP can distinguish different strokes steadily. Intuitively, we can apply two thresholds to classify these three basic strokes according to the absolute values of slopes in MNP series.

*5.3.3 Threshold Selection.* We now discuss how to determine the exact value of these thresholds in order to identify different strokes. To achieve this goal, the empirical experiments mentioned in Section 5.3.2 are conducted in three environments includes an empty hall, an office room and a living room. We collect 900 samples in total. A gradient search process is adopted on the collect data set from previous Section 5.3.2, and we determine the threshold when the global highest recognition accuracy is achieved from each type of strokes.

To verify whether the robustness of the thresholds in different environments. We conduct the empirical experiment in Section 5.3.2 in three different rooms(e.g., empty hall, office and living room). The layouts of later two rooms are shown in Figure 26. Figure 17 shows the accuracy for identifying strokes with determined thresholds in these rooms, with the overall confusion matrix shown in Figure 18. We can draw the conclusion that these thresholds are stable to identify strokes in different environments , and they can be directly applied to gesture recognition in different scenarios.

## 5.4 Recognize Gestures with MNP

This section presents the specific steps about how to recognize gestures using the MNP. As mentioned before, a gesture can be decomposed into temporal combination of strokes, and the combination can be represented by a

unique temporal identification sequence for each gesture. For any predefined gestures, we can generate their corresponding sequence using prior knowledge as a reference profile. Then for any input gesture, we only need to match the generated identification sequence with the reference and recognize it finally.

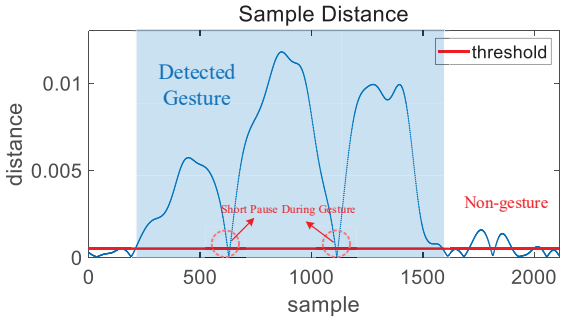



Fig. 19. Gesture is detected using the sample distance of CSI ratio

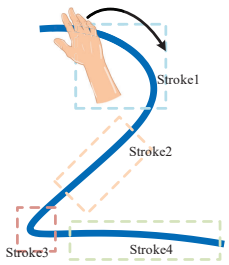



Fig. 20. Basic Strokes of '2'

*5.4.1 Gesture Detection.* Gestures are identified with the help of distance information of corresponding CSI ratio samples in the complex plane (as shown in Figure 7(a)). As shown in Section 2, when user's hand moves, the dynamic phase components of CSI will change, which led to the rotation of CSI ratio in the complex plane. At this time, the samples of CSI ratio will be dispersed on the plane. On the other hand, when users keep their hands still, the samples of CSI ratio will gather together in the plane. On this basis, we leverage the distance of CSI ratio samples to detect the hand movements.

We set an threshold on the distance to indicate whether the hand is stationary or moving. After system startup, we require a user to keep the hand still in the first 0.5 $s$ then slightly move the hand. The threshold is determined by comparing the corresponding sample distance information before and after the hand starts to move. Figure 19 shows the sample distance of the gesture '2' on one receiver. Generally the hand is moving when the distance is higher than the threshold. However the hand may have a very short pause during the gestures (usually happening in the corner stroke). Therefore a very short stationary segment (less than 20 samples with the sampling rate of 400 $Hz$) is also considered as the segment with moving hand (marked with the red circle). A gesture is detected when the hand moves for more than 1 second according to the sample distance. Therefore, we know the start and end time of each gesture. Then the MNP is extracted from the CSI in the corresponding time.

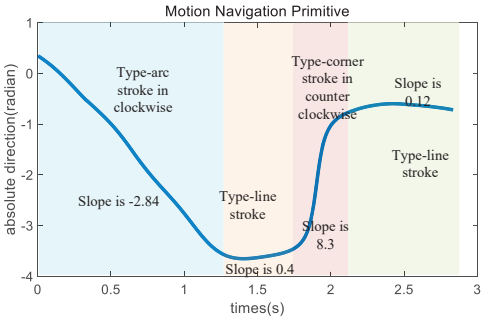



Fig. 21. Optimized MNP of Drawing '2'

| *Basic Strokes* | *Denotation* |
|---|---|
| removable line | $l^*$ |
| unremovable line | $l$ |
| clockwise&counterclockwise arc | $a_1$,$a_2$ |
| clockwise&counterclockwise corner | $c_1$,$c_2$ |

Fig. 22. Denotation of Strokes

*5.4.2 Strokes Segmentation.* We can segment the gesture into strokes using the absolute value of slope on the MNP. Figure 21 illustrates the MNP of digit '2'. Four segments with three different trends of the MNP curve are observed, which are correspond to four gesture strokes. We use two principles to segment strokes. The first is that the slopes within the same segments are similar. The second is that the slopes between adjacent segments are significantly different. Naturally, we marked the points where the slope suddenly changes as the segmentation points. By doing this, we segment a complicated gesture into several basic strokes.

*5.4.3 Strokes Identifying & Identification Sequence Extraction.* As described in Sec 5.3, despite different settings, the slopes of the MNP series are categorized into three groups which are matched with three basic strokes respectively. We apply the threshold-based strategy on these segments to distinguish which types of strokes they are. The two thresholds are selected as in Section 5.3.3. After that, we estimate the average of the absolute value of slopes for every MNP segment. Based on the absolute value of the slope of MNP, the gesture '2' is decomposed into four strokes as shown in Figure 21.

We still have one more step to generate the temporal identification sequence. As we describe in the Sec 5.2, the clockwise directionality of strokes can be determined by the trend of the MNP series. We only consider the type-corner and type-arc strokes. By simply determining whether the average slope of the corresponding segments is positive or negative, we can find whether the MNP increases or decreases, and determine whether the hand moves toward counterclockwise or clockwise.

With all of the information, we can extract the temporal identification sequence which reflects the order of basic strokes with their types and clockwise directionality. Thus the extracted sequence can also be described as *{type-arc in clockwise, type-line, type-corner in counterclockwise, type-line}*. For most gesture sets, their moving direction changes of every gesture are distinguishable. And the extracted temporal identification sequence can be a good indicator for recognizing these gestures.

*5.4.4 Reference Sequence Generation & Gesture Recognition.* For gesture sets with predefined tracks, the strokes each gesture contains can be determined by the prior knowledge in Sec.5.2 without sampling data in advance. When putting it into practice, before the recognition, a user is required to perform these gestures at first and extract the MNP to identify contained strokes through the previous mentioned steps. As such, we can generate a temporal identification sequence as the reference sequence for each gesture. This process only requires very few pre-collect samples as long as we can obtain the MNP series, and it only needs to be done once with gestures performed in one location and orientation. Therefore we can cut the labor of conducting a theoretical analysis of the hand movements.

After generating the reference sequence, the gesture can be recognized by matching the temporal identification sequence with the reference. However, user's performing styles can be different. In some rare cases, the user may perform a same gesture with slightly different moving trajectories, and their contained strokes may be also different. For example, let's assume the obtained reference sequence of '2' is *{type-arc in clockwise, type-line, type-corner in counterclockwise, type-line}*. When a user performs '2' , it is possible that the performed gesture may miss the second stroke. In order to cover more situations in practice, we have generated more reference sequences. We applied two rules for complicated gestures with three more strokes to generate the reference sequences. The first one is that the type-corner and type-arc strokes are usually critical and can not be removed. The second one is that the type-line stroke near a type-arc stroke with the same clockwise directionality can be removed. According to these rules, the first type-line stroke in the predicted sequence of '2' is removable. And we can generate another reference identification sequence of '2' which is *{type-arc in clockwise, type-corner in counterclockwise, type-line}*.

Without ambiguity, we use symbols to denote different kind of basic gesture strokes in the Figure 22. Thus the reference identification sequence of drawing '2' can be presented as $\{a_1\ l^*\ c_2\ l\}$. When an extracted temporal

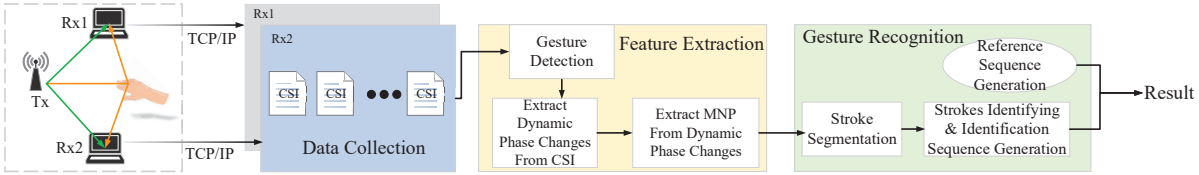



Fig. 23. System Architecture of WiGesture.

identification is successfully matched with one of all generated reference sequences of a predefined gesture, the input gesture is identified.

## 6 EVALUATION

To evaluate our proposed method, we design and implement a prototype system called WiGesture and conduct comprehensive experiments.

### 6.1 Overview of the Prototype System

Figure 23 provides an overview of the architecture of WiGesture. As illustrated, WiGesture is built on top of three Wi-Fi devices including one transmitter and two receivers. As shown in Figure 24, one of them is set as the transmitter (Tx) and the other two are set as the receivers (Rx). Each receiver is equipped with two omni-directional antennas. All of the devices are mounted on tripods and the direction of antennas is placed in parallel to the ground to better capture the motions of the hand. The system runs with MATLAB on a server with an i7-6700 CPU and 16G RAM collects CSI streams from two receivers. It takes 70 *ms* to process input data of one second. The CSI signals extracted from the two receivers are processed through a pipeline that consists of three modules: 1) *Data Collection Module*, 2) *MNP Series Extraction Module*, and 3) *Gesture Recognition Module*.

*6.1.1 Data Collection Module.* WiGesture collects CSI signals on receivers and sends them to a server through TCP/IP socket connection for processing. The CSI tool [7] is used to collect CSI measurements. The Tx sends CSI packets with a sampling rate of 400 Hz. The frequency of the Wi-Fi signal is set to the 5.24 GHz and the band-width is set to 40 MHz. Specifically, when the AP is broadcasting the Wi-Fi signals at a predefined channel, the two receivers are set on injection mode to generate CSI data streams. The default sampling rate is set to 400 $Hz$. For each antenna, we can get a CSI stream and every stream contains 30 sub-carriers.

Packet loss may occur on the receiver during signal transmission. WiGesture uses the timestamps in each CSI packet to synchronize two receivers. With default sampling rates we can calculate the expected time interval between each packet and mark the missing packets with a null time slot. After calculating the CSI ratio on each receiver, the value of these slots is obtained by interpolating adjacent data.

*6.1.2 Feature Extraction Module.* WiGesture first detect the gesture as illustrated in Sec 5.4.1. Then WiGesture extracts the corresponding MNP series to the gesture as described in Sec 4. Specifically, it firstly leverages the CSI ratio proposed in [38] to remove the phase offset and reduce the impulse noise in the CSI signals. For each pair of transceivers, WiGesture estimates the phase changes of the dynamic component of CSI. Lastly, the MNP is extracted by calculating the ratio of CSI's dynamic phase changes from two pairs of transceivers.

*6.1.3 Gesture Recognition Module.* As previously illustrated in the Sec 5.4, for each gesture, WiGesture firstly generates an identification sequence as the reference. For each input gesture in system running, WiGesture decomposes the gesture into several basic strokes and generates an identification sequence from the extracted

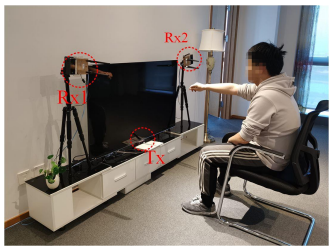



Fig. 24. Data Collection.

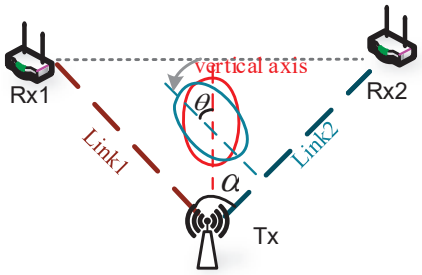



Fig. 25. Performing Gestures with different orientations

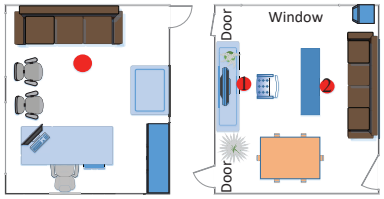



Fig. 26. Layouts of two indoor environments

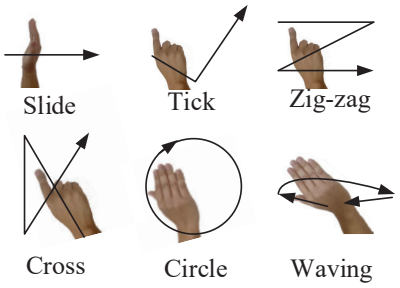



Fig. 27. Illustration of six basic gestures


Fig. 28. Illustration of four conducting gestures

MNP. By comparing the references with the generated identification sequence, WiGesture is able to recognize the performed gestures.

## 6.2 Experimental Setup

*Gesture Sets:* To demonstrate the generability of WiGesture, we evaluated WiGesture on three diverse sets of gestures. Gesture Set#1: a gesture set that covers six basic gestures as shown in Figure 27. Gesture Set#2: a gesture set that contains four conducting gestures as shown in Figure 28; and Gesture Set#3: a gesture set that covers ten digits as shown in Figure 10.

*Experimental Environment:* The system is evaluated in three environments includes a living room (4 m × 7 m), an office room (4 m × 5 m) and an empty hall (8 m × 12 m). Both the office room and the living room are multi-path rich environments with furniture as shown in Figure 26.

*Subjects:* We recruit 20 subjects in total including 13 males and 7 females aging from 21 to 39 years old. Four of them are the authors of this paper, the rest have no prior knowledge about our system.

*Data Collection:* We conducted the experiments for all the subjects performing gestures with different position-specific factors on three gesture sets. Each individual performed each gesture with one location&orientation in 10 times. A total of 3600, 2400, and 6000 data were collected in three sets. As shown in Figure 24, when collecting data, the subjects are required to sit in front of the transceivers. When performing gestures, they wave their hand in the air with their forearms and hands hovered and outstretched. For each gesture, the subject first keeps still, and starts performing gestures by the way we defined, and stops in the end without performing other activities.

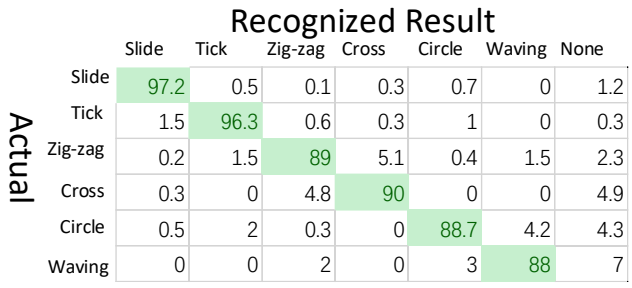



Fig. 29. Confusion matrix of Gesture Set#1.

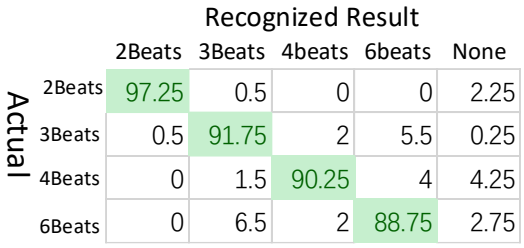



Fig. 30. Confusion matrix of Gesture Set#2.

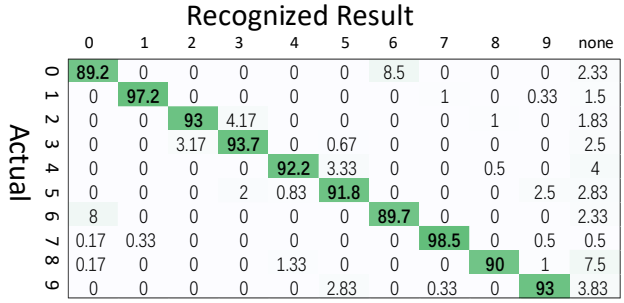



Fig. 31. Confusion matrix of Gesture Set#3.

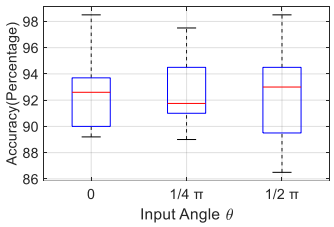



Fig. 32. Recognition Accuracy with Different Orientations

It should be noticed that during data collection, the subjects do not need to move the hand with the same initial location and precise track.

## 6.3 Overall Recognition Accuracy

Figure 29 shows the confusion matrix of Gesture Set#1. As the gestures in the set are relatively simple and only contain a few different strokes, WiGesture is able to easily recognize them with an average accuracy of 94.5%. Figure 30 shows the confusion matrix Gesture Set#2. The reference sequences can be predicted in denotation which are $\{lc_2l\}$,$\{lc_2la_1l\}$, $\{lc_1la_2la_1l^*\}$ and $\{la_2la_1\}$ for two beats, three beats, four beats, and six beats. Although the gestures in this set are more complicated, WiGesture is still able to achieve an average recognition accuracy of 92%. Figure 31 shows the confusion matrix of Gesture Set#3. The average of overall recognition accuracy is 92.8%. At the same time, we observe that there is some confusion with the recognition results of digit '0' and '6'. This is because '6' contains only one more type-line stroke than digit '0', and subjects did not perform the type-line stroke for enough time under some circumstances.

In the following, we evaluate the position-independent sensing capability (§5.3), robustness to real-world factors (§5.4) of WiGesture and compare it with state-of-the-arts (§5.5). We choose Gesture Set#3 for those evaluations. We choose this gesture set is that it contains enough number of both simple and complicated gestures. Figure 13 shows the corresponding MNP series. The predicted identification sequences for each gesture are shown in Table 1.

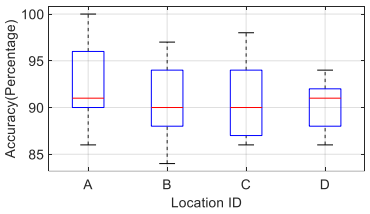



Fig. 33. Different Performing Locations

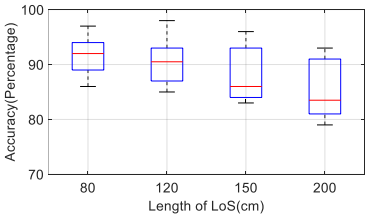



Fig. 34. Different Length of Line-of-Sights

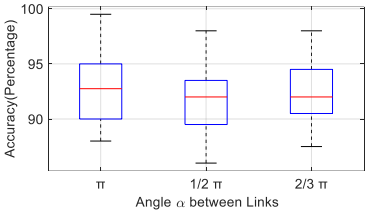



Fig. 35. Different Angles between Links

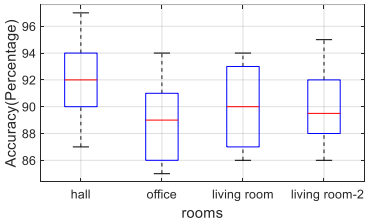



Fig. 36. Different Environments

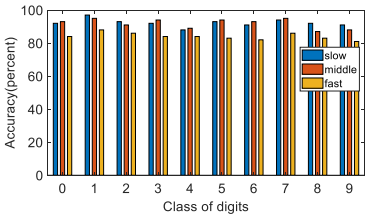



Fig. 37. Impact of Moving Speed

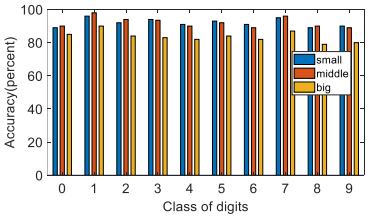



Fig. 38. Impact of Drawing Size

## 6.4 Evaluation of Position-independent Sensing Capability:

*6.4.1 Different Orientations of Gestures:* As shown in Figure 25, gestures are performed in different orientations. We use the input angle of the gesture to specifically represent the orientations. The first input angle $\theta$ is marked as 0, as the vertical axis of a digit is orthogonal to the connecting line of two receivers. And the $\theta$ indicates the angle that the vertical axis of the digit rotates counterclockwise. The second and third input angles are $1/4\pi$ and $1/2\pi$ respectively. All 10 digits are performed 50 times each. Figure 32 shows the box plot of accuracy. We can observe the high consistency across all the scenarios which proves the ability of WiGesture to recognize gestures with different orientations.

*6.4.2 Different Locations of Gestures:* We evaluate WiGesture by performing gestures at four different locations shown in Figure 11. The LoS is set to $150cm$. The locations A and D are closer to the devices with a distance within $20cm$. Location B is close to the link line between two Rxs and location C is at least $40cm$ from that link line. In each location, each digit is performed 50 times. Figure 33 shows the results. We can see that WiGesture can recognize gestures performed at different locations. However, the accuracy of location A and D is lower. It can be explained by a previous study [39] that when the reflected object is close to the LoS, the signal can also be affected by the diffraction besides reflection. Our CSI-ratio model-based method may fail to extract dynamic phase changes to generate an accurate MNP profile.

Table 1. Reference Identification Sequences for 10 digits

| Digits | Reference Sequence |
|---|---|
| 0 | $\{a_2\}$ |
| 1 | $\{l\}$ |
| 2 | $\{a_1\ l^*\ c_2\ l\}$ |
| 3 | $\{a_1\ l^*\ c_2\ l^*\ a_1\}$ |
| 4 | $\{l\ c_1\ l\ c_1\ l\}$ |
| 5 | $\{l\ c_2\ l\ c_2\ l^*\ a_1\}$ |
| 6 | $\{1\ a_2\}$ |
| 7 | $\{1\ c_1\ l\}$ |
| 8 | $\{a_2\ a_1\}$ |
| 9 | $\{a_2\ c_1\ l\}$ |

Table 2. The digits recognition accuracy of 20 subjects (in percentage)

| Subjects | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No.1 | 90 | 100 | 96.7 | 93.3 | 93.3 | 96.7 | 86.7 | 100 | 90 | 93.3 | 94 |
| No.2 | 93.3 | 100 | 93.3 | 90 | 90 | 90 | 93.3 | 100 | 90 | 100 | 94 |
| No.3 | 86.7 | 100 | 90 | 93.3 | 90 | 93.3 | 90 | 100 | 93.3 | 90 | 92.6 |
| No.4 | 83.3 | 93.3 | 90 | 86.7 | 76.7 | 90 | 80 | 100 | 80 | 86.7 | 86.7 |
| No.5 | 96.7 | 96.7 | 93.3 | 96.7 | 90 | 93.3 | 96.7 | 96.7 | 86.7 | 100 | 94.7 |
| No.6 | 96.7 | 100 | 93.3 | 100 | 96.7 | 96.7 | 93.3 | 100 | 100 | 96.7 | 92.5 |
| No.7 | 93.3 | 100 | 96.7 | 100 | 93.3 | 96.7 | 90 | 100 | 93.3 | 93.3 | 97.3 |
| No.8 | 86.7 | 96.7 | 90 | 86.7 | 100 | 93.3 | 93.3 | 96.7 | 90 | 100 | 95.6 |
| No.9 | 90 | 96.7 | 86.7 | 93.3 | 90 | 100 | 90 | 100 | 93.3 | 93.3 | 93.3 |
| No.10 | 83.3 | 93.3 | 96.7 | 96.7 | 93.3 | 90 | 93.3 | 100 | 86.7 | 93.3 | 92.7 |
| No.11 | 86.7 | 93.3 | 90 | 90 | 90 | 80 | 83.3 | 96.7 | 80 | 96.7 | 88.7 |
| No.12 | 90 | 93.3 | 86.7 | 86.7 | 83.3 | 83.3 | 90 | 100 | 80 | 86.7 | 88 |
| No.13 | 90 | 100 | 96.7 | 100 | 96.7 | 90 | 80 | 100 | 93.3 | 86.7 | 93.3 |
| No.14 | 93.3 | 100 | 93.3 | 96.7 | 100 | 86.7 | 93.3 | 100 | 96.7 | 93.3 | 95.3 |
| No.15 | 90 | 96.7 | 96.7 | 100 | 93.3 | 93.3 | 96.7 | 93.3 | 86.7 | 90 | 93.7 |
| No.16 | 86.7 | 93.3 | 93.3 | 90 | 96.7 | 93.3 | 90 | 100 | 93.3 | 90 | 92.7 |
| No.17 | 90 | 96.7 | 93.3 | 93.3 | 93.3 | 96.7 | 93.3 | 100 | 90 | 100 | 94.7 |
| No.18 | 90 | 100 | 100 | 100 | 96.7 | 93.3 | 96.7 | 96.7 | 93.3 | 90 | 95.7 |
| No.19 | 86.7 | 100 | 93.3 | 90 | 93.3 | 90 | 86.7 | 100 | 96.7 | 93.3 | 93 |
| No.20 | 80 | 93.3 | 90 | 90 | 86.7 | 90 | 76.7 | 90 | 86.7 | 86.7 | 87 |
| Avg | 89.2 | 97.2 | 93 | 93.7 | 92.2 | 91.8 | 89.7 | 98.5 | 90 | 93 | 92.8 |

*6.4.3 Different LoS of Transceivers:* We evaluate WiGesture with four different LoS between the transmitter and the receiver, which were 80$cm$, 120$cm$,150$cm$ and 200$cm$. And a user is asked to perform 10 digits 50 times in each setting. The result is shown in Figure 34. The accuracy of short LoS is slightly better than that of long LoS. This is probably because a short LoS distance can result in a stronger reflection signal of the moving hand, which is more robust to the multi-path environment.

*6.4.4 Different Angles between Links of Transceivers:* The angles between two links are marked as $\alpha$ in Figure 25. Specifically, the $\alpha$ is set to $1/3\pi$,$1/2\pi$ and $2/3\pi$ respectively. WiGesture still achieves high consistent accuracy across all different $\alpha$. Figure 35 shows the related result. Based on this result, we believe that our system requires less effort on deployment as the devices do not need to be placed at precise locations.

*6.4.5 Different Subjects:* We examine the impact of different subjects on the performance of gesture recognition on our testing set. Tab. 2 shows the detail of recognition accuracy for all subjects involved. The results are shown in percentages.

*6.4.6 Different Wireless Environments:* We deploy WiGesture in three different rooms including an empty hall, an office room and a living room. The last two are rich multi-path environments. The red dots in Figure 26 mark where we place our devices in the room. Specifically, 4 subjects were involved in the experiments with the $\theta$ of 0 and $\alpha$ of $1/2\pi$, and each digit in the gesture set was performed 25 times. Specifically, the user first performs gestures at location #1. Then we examine the impact of the changing environment with the changed layout of the living room by closing the window and moving the table next to it, where devices are deployed at location #2. The overall accuracy is shown in Figure 36, in which "living room-2" denotes the living room with a changed layout.

Our result shows that the accuracy in the office room and living room is a little bit lower but still promising, while the accuracy in the changed living room is stable. Although how multi-paths superimposed is irrelevant to our method as we only focus on the dynamic reflection signals. But a rich multi-path environment may also result in an unstable combination of static and dynamic components on CSI signals, and slightly distort the

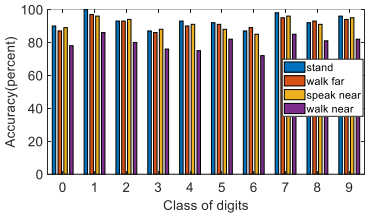



Fig. 39. Impact of Ambient Motion

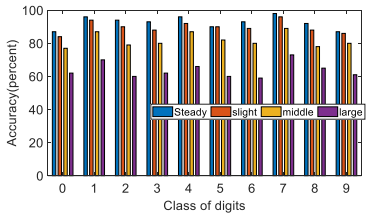



Fig. 40. Impact of Body Movement

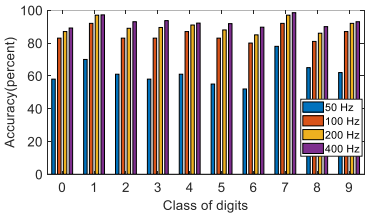



Fig. 41. Impact of Sampling Rate

extracted dynamic phase changes. Even under such an environment, the steps we take in Sec 4.2 greatly reduce the influence which makes WiGesture more robust to different wireless environments.

## 6.5 System Robustness to Various Real-World Factors

*6.5.1 Impact of Motion Speed:* We evaluate WiGesture with three motion speed of the hand. Four subjects participate in this experiment in the meeting room and everyone performs each gesture 25 times for every speed. Specially, we hand a tablet to catch the moving track and time as the ground truth and estimate the average motion speed. These three motion speeds are 3.5cm/s,6.5cm/s and 9cm/s respectively. Figure 37 shows the overall accuracy of recognizing 10 gestures with different motion speeds. All the results at three speeds are above 85% which is acceptable. The result at the fast speed is a little worse than others. We speculate that when moving fast, it is more difficult for subjects to drawing every stroke the pre-defined gesture is supposed to have.

*6.5.2 Impact of Sizes of Moving Track:* We evaluate WiGesture with different sizes of the moving track for the gestures. We add a big and small size compared with the default one in previous experiments. Subjects draw the digits ensuring the tracks are within the rectangles of 8 cm × 8 cm, 15 cm × 15 cm and 20 cm × 20 cm on the board hanging in front of them. Figure 38 shows the recognition accuracy for different sizes. We find there is no difference of the scenarios in the small and middle sizes. The accuracy decreases a little for the big size. It is mainly because the subject's hand is more likely to be unstable when drawing a long stroke.

*6.5.3 Impact of Ambient Motion:* We evaluate WiGesture with the ambient motion, i.e., the motion from others in the environment. The experiment is conducted in the living room. The user performs gestures with two other subjects standing one meter away from the devices. We require one of them to act in the room. The subject first walks close to the door, which is far from the devices (more than 4 meters), then speaks near the devices (within 2 meters). At last, the subject walks around the devices (within 2 meters). Each gesture is performed 100 times in each case. Figure 39 shows the performance of different scenarios. We find that the obvious human activities like walking near the device (within 3 meters) can severely distort the performance, but small activities like breathing or speaking have no noticeable influence.

*6.5.4 Impact of Body Movements:* We evaluate the impact of body movements on WiGesture. The users perform gestures in four different ways, i.e., steady, slight, middle and large movements, with moving hand at the speed around 6.5cm/s. Each gesture is performed 10 times and the size of the track is around 20x20 *cm*. Specifically, they keep their forearms and hands hovered and outstretched. The motion of the hand should be steady and smooth. And users try to keep the rest of the body still. For the slight moving, users slightly shake their hand every second, with the amplitude around 1*cm*. For the middle moving case, users perform gestures shaking hands the same as the slight-moving case. Meanwhile, they shake their body back and forth every 2 seconds, with the moving range of the chest around 6 *cm*. For the large moving case, users shake hands every second with an amplitude of around 3 *cm*. Similarly, they shake the body back and forth every 2 seconds, with the moving range

of the chest around 12*cm*. It is worth noting that although user's motion ranges can not be the same every time, the differences within each type of body movement are much smaller than that between different types.

Figure 40 shows the results. The normal body movements may lead to little degradation of performance. However, WiGesture still achieves an accuracy with an average above 80% in the third case. It can be explained that as the result of the polarization of antennas, WiGesture is mainly sensitive to the plane which is perpendicular to the antenna extension direction. Most body movements are not in the sensitive area as the user sits in front of the plane as shown in Figure 24. Moreover, the algorithm we applied to extract MNP also reduces the impact of body movements.

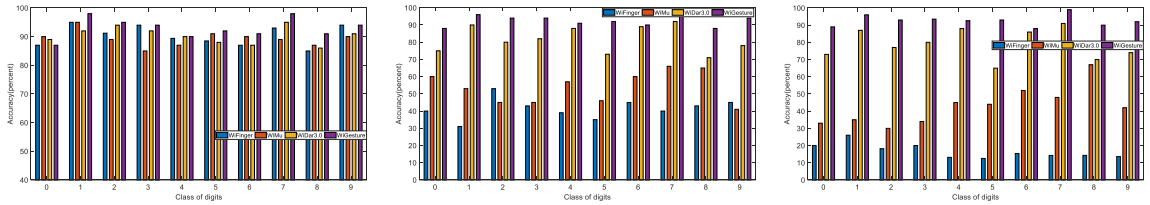
*6.5.5 Impact of Lower Sampling Rate:* In most of our experiments, the sampling rate is 400 Hz. However, in practice, the sampling rate can be much lower. Thus we evaluate WiGesture with different sampling rates, i.e.,25$Hz$, 50 $Hz$, 100 $Hz$, 200 $Hz$, 400 $Hz$. For each scenario, the user is asked to perform all digits 10 times. The results are shown in Figure 41. We can find that the higher the sampling rate is, the higher accuracy will be. When the sampling rate is higher than 50 $Hz$, the decreasing of the accuracy is not significant. However, when the sampling rate is lower than 50 $Hz$, it will decrease sharply.

## 6.6 Comparison with State-of-the-Arts

Finally, we compare WiGesture with state-of-the-art Wi-Fi-based gesture recognition systems including WiFinger [20], WiMu [21], and WiDar3.0 [44]. Specifically, we conduct the comparison in three scenarios considering different device deployments and gesture performing styles. The digit set is selected as the test set of our experiments.

*6.6.1 Implementation of State-of-the-Arts.* Specifically, WiFinger [20] uses time-domain waveform shapes to recognize pre-defined gestures. To evaluate this method, we apply Dynamic Time Warping (DTW) to align and compare the waveform with reference profiles. We randomly select one-tenth of the dataset to build the reference profiles. WiMu [21] uses the patterns of frequency distribution to recognize gestures. We apply Fast Fourier Transform on our dataset to extract the frequency spectrum. And we down-sample the spectrum to a 65-sample vector. We apply Radial Basis Function kernel SVM to classify the digits. In each scenario, there are at least 300 samples for each digit and we select 10 sub-carriers for each data, thus the dataset contains more than 30000 data samples. We randomly select one-fifth of the dataset to train the SVM model and predict the result with the rest. WiDar3.0[44] utilizes DFS profiles from at least 3 links to generate BVP to recognize each gesture. Thus we add an extra receiver in our experiments and apply WiDar3.0[44] in our dataset. Specifically, we randomly pick one-fifth of the experiment set which equally contains all different orientations as training samples, and test the model using the rest data. We also record the initial positions of each data which is also required by WiDar3.0[44].

*6.6.2 Scenario A: Fixed Deployment and Controlled Gesture.* First, we compare WiGesture with other State-of-the-Arts with fixed deployment and controlled gesture. This means that the deployments of devices are kept unchanged. The initial position, the size and the orientation of the gesture are also kept unchanged for each sample. In other words, for the same gesture, the relative position between the hand and the device does not change. Specifically, the length of Los is set to 120 *cm* and the angle between two links is set to 1/2 $\pi$. Each type of gesture is performed 300 times. The initial position of gesture is location B shown in Figure 11, the input angle of the orientation is 0 $\pi$ with the controlled size around 15 cm $\times$ 15 cm. The comparison results are shown in Figure 42(a). In this position-specific scenario, the patterns in the CSI signal are similar. Thus WiFinger [20], WiMu [21] can achieve good performance. As motions of the same gestures are very similar, WiDar3.0 [44] can also obtain a consistent velocity profile to achieve high accuracy.




(a) Scenario A (b) Scenario B (c) Scenario C

Fig. 42. The Comparison Results with Different Techniques under Three Scenarios

*6.6.3 Scenario B: Fixed Deployment and Uncontrolled Gesture.* Second, we compare WiGesture with other State-of-the-Arts with fixed deployment and uncontrolled gesture. The deployment of the devices is the same as in Scenario A. However, we do not require subjects to perform the gesture with a precise moving track. Each type of digits is performed with the same orientations as in Sec. 6.4. Every orientation contains 100 data. We also do not require a specific initial location. Considering different orientations, the initial locations of the gestures are not the same. Thus for the same gesture, the relative positions between the hand and the device can be different. The comparison results are shown in Figure 42(b). We can see that WiGesture and WiDar3.0 [44] outperform the others. It This is because that the pattern-based approaches like WiFinger [20] and WiMu [21] are highly sensitive to locations, orientations as well as other position-specific factors. They are not capable of support position-independent sensing.

*6.6.4 Scenario C: Unfixed Deployment and Uncontrolled Gesture.* Lastly, we conduct the experiment while the deployment of devices is unfixed and the gesture is uncontrolled. We set three different lengths of LoS which are 80 cm, 120 cm and 150 cm respectively. There are also three different angles between two links which are 0 $\pi$, 1/4 $\pi$ and 1/2 $\pi$. Combining them together there are 9 different deployments. Users can choose any one of the four initial locations shown in Figure 11. They can perform gestures with any orientations they like and don't have to follow a precise moving track. In each deployment, each type of gesture is performed 100 times. As shown in Figure 42(c), WiGesture achieves better performance than state-of-the-arts. It outperforms WiFinger [20], WiMu [21], and WiDar3.0 [44] by 75.7%, 49.6%, and 13.5%, respectively. Like the previous experiment, WiFinger [20] and WiMu [21] achieve a relatively poor performance. Meanwhile, we find that WiGesture is better than WiDar3.0 [44] in both scenario B and scenario C. It is because the gestures we performed are relatively micro (15x15$cm$), with just three receivers deployed, WiDar3.0 [20] may not extract enough prominent DFS to generate consistent and robust velocity profiles for the same gesture in different locations$ orientations. In contrast, by utilizing the CSI ratio model to extract the dynamic phase changes of the CSI, WiGesture is able to achieve accurate gesture recognition with only two receivers.

## 7 DISCUSSIONS

### 7.1 Capability of MNP

MNP is capable of recognizing gestures with unique and distinguishable patterns of moving direction changes. However, if two gestures in the gesture set share similar patterns in moving direction changes, MNP may fail to recognize them. Nevertheless, we can extract more features (such as the moving speed changes) using the proposed hand-oriented view-based sensing strategy to recognize more gestures. The idea of the proposed sensing strategy is general and paves a new way for building practical Wi-Fi sensing systems.

### 7.2 Applications beyond the Hand-oriented View-based Sensing Strategy

In this paper, we build a prototype system called WiGesture to verify the effectiveness of the proposed sensing strategy. The technique based on our strategy only requires two pairs of transceivers to recognize gestures with different locations and orientations. The deployment of the device does not need to be fixed and there is no need to acquire the exact location of each device. In the future, considering widely deployed RF devices (such as smartphones and routers) in the smart room scenario, the proposed sensing strategy can reuse them as transceivers. Given all the advantages, we can support more practical applications which recognize gestures in the real world without the need to relocate these devices or to know their exact locations.

## 8 RELATED WORK

Contactless gesture recognition can be conceptually grouped into two categories of approaches: non-WiFi-based approaches and WiFi-based ones. Non-WiFi-based gesture recognition in general needs dedicated devices such as cameras[5, 19], RFID tags [23, 27], radars [4, 6, 16, 42, 43], and sonar devices [11, 31, 36]. In contrast, WiFi-based gesture recognition [2, 3, 12, 14, 18, 30, 32] senses gestures by leveraging the ubiquitous Wi-Fi signals transmitted from commodity Wi-Fi devices.

### 8.1 Non-WiFi-based Gesture Recognition

Computer vision-based gesture recognition systems [5, 19] use infrared cameras and LEDs to capture hand moving images. Although they work pretty well in line-of-sight (LoS) scenarios, their performance degrades significantly if the target is out of LoS or in the poor illumination scenario.

Radar-based gesture recognition systems can achieve high accuracy, but require specialized hardware whose cost is relatively high. For example, Wang et al. [27] designed RF-IDraw to track finger movements with RFID tags using multiple antenna arrays. RF-finger [23] implemented contactless finger tracking through tag arrays with only one RFID antenna. WiSee [16] utilized USRP with OFDM modulated signals to extract the Doppler shift of the finger movement. Google Soli[6] can track micro finger movements through constructing a Doppler profile using 60 GHz radar signals.

In terms of sonar-based gesture recognition systems, LLAP [31] utilizes the phase change of the continuous wave sound signal for finger movement tracking, and FingerIO [11] transmits specially modulated OFDM signals and locates finger according to the change of echo profiles. Strata [36] combines the frame-based approach and the phase-based approach. However, sonar-based systems often suffer from environmental noise and privacy concerns (i.e., unattended recording of voice).

### 8.2 WiFi-based Gesture Recognition

WiFi-based gesture recognition systems can also be divided into two categories according to whether they can guarantee performance when recognizing gestures across different positions.

For systems that do not support position-independent gesture recognition, WiGest [1], WiMU [21], Mudra [41] have claimed their contribution on motion recognition when there is no significant difference in the environment. Though there exist various features extracted from CSI signals, most of the work concentrate on the statistics from waveform and frequency. The key insight behind is to find how the signal information reflects the motion profile and the fine-grained profile can be extracted to enable fine-motion recognition. Therefore, with coarse CSI profile, E-eyes [10] mainly solve the activity recognition and make the best use of in-place information. WiGest[1] obtain the frequency distributions using Discrete Waveform Transform and get the ability to recognize hand gestures. WiMu [21] further explores the information provided by CSI and apply Short Time Fourier Transform (STFT) to generate gesture profile. WiFinger[20] calculate the patterns in the time domain using principal component identification to identify hand gestures. WiTrace [28] leverages OFDM signals to track hand

movements at centimeter-level. Mudra [41] find out the relationship between two antennas that are sensitive to the hand movement direction. Above all, the relationship between signal and hand gesture is complicated and can be affected by plenty of factors, which prevent them from solving the position-dependency problem.

For systems that support position-independent gesture recognition, WiAG [22] and WiDar3.0 [44] strengthen the ability to solve the position-dependency problem. WiAG [22] analyzes how the hand moves relative to the transceivers to develop a translation function. This function is designed to generate a virtual sample for gestures in possible domains. WiDar3.0 [44] uses velocity profiles of gestures extracted from Doppler Frequency Shift (DFS) on at least three links, which acts as unique indicators of gestures for recognition. However, in order to recognize gestures in different locations and orientations, they still require position-related information such as the device deployment locations, initial locations and orientations of the hand. In other words, they recognize gestures in fixed deployments, which makes the inconvenience in practice.

## 9 CONCLUSION

In this paper, we present a novel gesture recognition strategy to relieve the fundamental position-dependency problem. Specifically, we shift our observation from the traditional transceiver view to the hand-oriented view and extract a novel position-independent feature named Motion Navigation Primitive (MNP). MNP exploits the moving direction of the hand movement, shares consistent patterns when users perform gestures with different locations and orientations. Using the proposed MNP, we develop a technique that decomposes hand movement trajectory into a combination of basic gesture strokes, which makes gestures easy to be identified. To validate our sensing strategy and extracted feature, we implement a gesture recognition prototype system named WiGeture. The results show that WiGesture achieves an overall recognition accuracy of more than 90% in three different gesture sets, and is robust to different domains and real-world factors.

Given its superiority, we believe the proposed method opens a new door towards the practical deployment of wireless human sensing systems in the real world. The idea proposed in this paper may shed light on a promising way to solve the position-dependency problem in other WiFi sensing applications.

## ACKNOWLEDGMENTS

This research is supported by NSFC A3 Project 62061146001, PKU-Baidu Funded Project 2019BD005, PKU-NTU collaboration Project.

## REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. WiGest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 1472–1480.

[2] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using wifi signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 90–102.

[3] Xiaoxiao Cao, Bing Chen, and Yanchao Zhao. 2016. Wi-Wri: Fine-grained writing recognition using Wi-Fi signals. In *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 1366–1373.

[4] Biyi Fang, Nicholas D Lane, Mi Zhang, Aidan Boran, and Fahim Kawsar. 2016. BodyScan: A Wearable Device for Contact-less Radio-based Sensing of Body-related Activities. In *14th ACM Conference on Mobile Systems, Applications, and Services (MobiSys' 16). ACM.*

[5] Makiko Funasaka, Yu Ishikawa, Masami Takata, and Kazuki Joe. 2015. Sign language recognition using leap motion controller. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, 263.

[6] Google. [n.d.]. Project Soli. https://www.youtube.com/watch?v=0QNiZfSsPc0. Accessed Jan 15, 2020.

[7] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review* 41, 1 (2011), 53–53.

[8] Francis A Jenkins and Harvey E White. 1937. *Fundamentals of optics*. Tata McGraw-Hill Education.

[9] Shengjie Li, Xiang Li, Qin Lv, Guiyu Tian, and Daqing Zhang. 2018. WiFit: Ubiquitous Bodyweight Exercise Monitoring with Commodity Wi-Fi Devices. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing,*

*Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 530–537.

[10] Zhengxiong Li, Zhuolin Yang, Chen Song, Changzhi Li, Zhengyu Peng, and Wenyao Xu. 2018. E-Eye: Hidden electronics recognition through mmwave nonlinear effects. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 68–81.

[11] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1515–1525.

[12] Kai Niu, Fusang Zhang, Yuhang Jiang, Jie Xiong, Qin Lv, Youwei Zeng, and Daqing Zhang. 2019. WiMorse: A Contactless Morse Code Text Input System Using Ambient WiFi Signals. *IEEE Internet of Things Journal* 6, 6 (2019), 9993–10008.

[13] Kai Niu, Fusang Zhang, Xuanzhi Wang, Qin Lv, Haitong Luo, and Daqing Zhang. 2021. Understanding WiFi Signal Frequency Features for Position-Independent Gesture Sensing. *IEEE Transactions on Mobile Computing* (2021), 1–1. https://doi.org/10.1109/TMC.2021.3063135

[14] Kai Niu, Fusang Zhang, Jie Xiong, Xiang Li, Enze Yi, and Daqing Zhang. 2018. Boosting fine-grained activity sensing by embracing wireless multipath effects. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. ACM, 139–151.

[15] Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. 2018. FallDeFi: Ubiquitous Fall Detection using Commodity Wi-Fi Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 155.

[16] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 27–38.

[17] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 6.

[18] Muhammad Shahzad and Shaohu Zhang. 2018. Augmenting user identification with WiFi based gesture recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–27.

[19] Chao Sun, Tianzhu Zhang, and Changsheng Xu. 2015. Latent support vector machine modeling for sign language recognition with Kinect. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 2 (2015), 20.

[20] Sheng Tan and Jie Yang. 2016. 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*. ACM, 201–210.

[21] Raghav H Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-User Gesture Recognition Using WiFi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 401–413.

[22] Aditya Virmani and Muhammad Shahzad. 2017. Position and orientation agnostic gesture recognition using wifi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 252–264.

[23] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. 2018. Multi-Touch in the Air: Device-Free Finger Tracking and Gesture Recognition via COTS RFID. In *Proceedings of the Conference on Computer Communications (INFOCOM)*. IEEE, 1691–1699.

[24] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human respiration detection with commodity wifi devices: do user location and body orientation matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 25–36.

[25] Hao Wang, Daqing Zhang, Kai Niu, Qin Lv, Yuanhuai Liu, Dan Wu, Ruiyang Gao, and Bing Xie. 2017. MFDL: A Multicarrier Fresnel Penetration Model based Device-Free Localization System leveraging Commodity Wi-Fi Cards. *arXiv preprint arXiv:1707.07514* (2017).

[26] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. 2017. RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices. *IEEE Transactions on Mobile Computing* 16, 2 (2017), 511–526.

[27] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: virtual touch screen in the air using RF signals. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 235–246.

[28] Lei Wang, Ke Sun, Haipeng Dai, Wei Wang, Kang Huang, Alex Liu, Xiaoyu Wang, and Qing Gu. 2019. WiTrace: Centimeter-Level Passive Gesture Tracking Using OFDM signals. *IEEE Transactions on Mobile Computing* (2019).

[29] Wei Wang, Alex X Liu, and Muhammad Shahzad. 2016. Gait recognition using wifi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 363–373.

[30] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*. ACM, 65–76.

[31] Wei Wang, Alex X Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 82–94.

[32] Dan Wu, RuiYang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. 2020. FingerDraw: Sub-wavelength Level Finger Motion Tracking with WiFi Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol* 1, 1 (2020).

[33] Dan Wu, Daqing Zhang, Chenren Xu, Hao Wang, and Xiang Li. 2017. Device-Free WiFi Human Sensing: From Pattern-Based to Model-Based Approaches. *IEEE Communications Magazine* 55, 10 (2017), 91–97.

[34] Dan Wu, Daqing Zhang, Chenren Xu, Yasha Wang, and Hao Wang. 2016. WiDir: walking direction estimation using wireless signals. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 351–362.
[35] Nan Yu, Wei Wang, Alex X Liu, and Lingtao Kong. 2018. QGesture: Quantifying Gesture Distance and Direction with WiFi Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 51.
[36] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 15–28.
[37] Youwei Zeng, Dan Wu, Ruiyang Gao, Tao Gu, and Daqing Zhang. 2018. FullBreathe: Full Human Respiration Detection Exploiting Complementarity of CSI Phase and Amplitude of WiFi Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 148.
[38] Youwei Zeng, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. 2019. Farsense: Pushing the range limit of wifi-based respiration sensing with csi ratio of two antennas. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–26.
[39] Daqing Zhang, Hao Wang, and Dan Wu. 2017. Toward centimeter-scale human activity sensing with Wi-Fi signals. *Computer* 50, 1 (2017), 48–57.
[40] Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. 2018. From Fresnel Diffraction Model to Fine-grained Human Respiration Sensing with Commodity Wi-Fi Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 53.
[41] Ouyang Zhang and Kannan Srinivasan. 2016. Mudra: User-friendly Fine-grained Gesture Recognition using WiFi Signals. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 83–96.
[42] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7356–7365.
[43] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 267–281.
[44] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 313–325.

# A APPENDIX

## A.1 Mathematical Foundation of MNP

In this subsection, we give the mathematical foundation to justify that the relative direction changes can be approximately represented by the ratio of two non-parallel velocity components.

As shown in Figure 43, we briefly denotes the velocity of the hand and its two velocity components as $v$,$v_a$ and $v_b$. Specifically, we mark the angle between $v$ and one of its velocity $v_a$ as $\theta$. The changes of $\theta$ represent the changes of moving directions as we keep the direction of $v_a$ constant during gesturing.

Now we justify that the ratio of two velocity components is corresponding to $\theta$. Through the geometry analysis, $\theta$ can be represented by the following equation:

$$\theta = arctan\frac{\frac{|v_b|}{|v_a|} - \cos\alpha}{\sin\alpha} \tag{12}$$

where $\alpha \ni [0, \pi]$ and is the constant angle of two velocity components. Clearly we can conclude that $\theta$ and $\frac{|v_b|}{|v_a|}$ is positive correlated. It means that $\frac{|v_b|}{|v_a|}$ solely depends on how the relative moving direction changes. Thus hypothetically it is feasible to roughly estimate how much the moving direction changed with the ratio of two velocity components.

## A.2 Verification of Sensing Hand Movement Using CSI Ratio Model

In order to show whether the CSI ratio model can be used to capture hand movements, we have conducted the following experiment. After carefully measuring the LoS length of the transceiver, we set different specific tracks

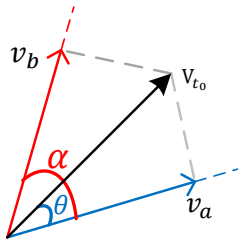



Fig. 43. Geometric illustration of MNP.

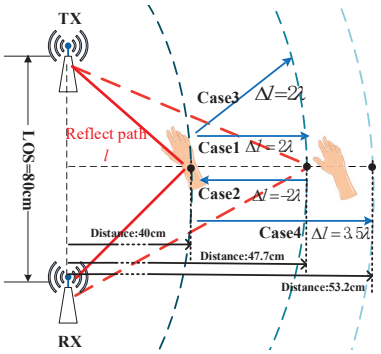
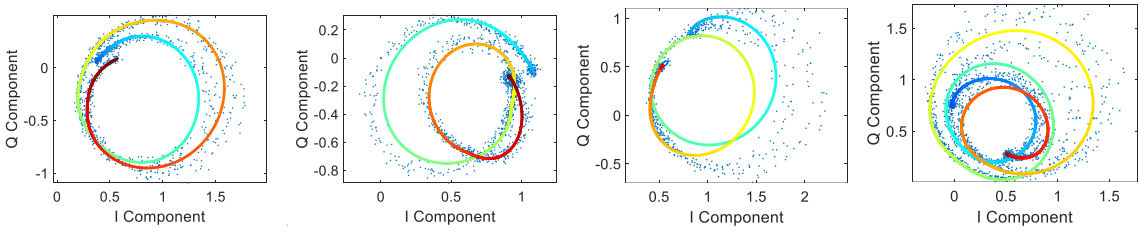



Fig. 44. Conceptual illustration of the Model Verification




(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Fig. 45. Demonstration of CSI ratio in different Cases

for hand movement and preset the change of reflection path as shown in Figure 44. We denote the wavelength of the signal as $\lambda$ and performed hand motions in four different cases: in Case1 & 2, the hand moves in two opposite direction but with the same expected reflect path changes around $2\lambda$; in Case 3, the hand moves in a different angle with the same path changes around $2\lambda$; in Case 4, the hand moves longer with the path changes around $3.5\lambda$.

Figure 45 illustrates the corresponding origin and the de-noised CSI ratio samples in these four cases. The sequence order of the de-noised samples is painted from dark blue to light red. In Figure 45(a) 45(c) and 45(d), CSI ratio rotates clockwise as reflect path increases, while in Figure 45(b), CSI ratio rotates counter-clockwise as reflect path decreases. We can observe that in Case 1, 2 and 3, the CSI ratio rotates around 2 rounds in the first three sub-figures as reflect path changes around $2\lambda$. The CSI ratio rotates more than 3 rounds and less than 4 rounds in Figure 45(d), while the reflect path increases in $3.5\lambda$. Therefore, the CSI ratio model is capable to sense the hand motions. With tolerable inaccuracy, the dynamic phase of CSI can be roughly expressed by the rotated angle $\varphi$ of the CSI ratio.