

# Mudra: User-friendly Fine-grained Gesture Recognition using WiFi Signals

Ouyang Zhang and Kannan Srinivasan  
Department of Computer Science and Engineering  
The Ohio State University, Columbus, OH 43210  
zhang.4746@buckeyemail.osu.edu, kannan@cse.ohio-state.edu

## Abstract

There has been a great interest in recognizing gestures using wireless communication signals. We are motivated in detecting extremely fine, subtle finger gestures with WiFi signals. We envision this technology to find applications in finger-gesture control, disabled-friendly devices, physical therapy etc. The requirements of mm-level sensitivity and user-friendly feature using existing WiFi signals pose great challenges. Here, we present Mudra, a fine-grained finger gesture recognition system which leverages WiFi signals to enable a near-human-to-machine interaction with finger motion.

Mudra uses a two-antenna receiver to detect and recognize finger gesture. It uses the signals received from one antenna to cancel the signal from the other. This “cancellation” is extremely sensitive to and enables us detect small variation in channel due to finger movements. Since Mudra decodes gestures with existing WiFi transmissions, Mudra enables gesture recognition without sacrificing WiFi transmission opportunities. Besides, Mudra is user-friendly with no need of user training. To demonstrate Mudra, we implement prototype on the NI-based SDR platform and use COTS WiFi adapter. We evaluate Mudra in a typical office environment. The results show that our system can achieve 96% accuracy.

## Keywords

Gesture Recognition; WiFi Signals; Signal Cancellation

## 1. INTRODUCTION

There is recent advances in pushing gesture recognition techniques. Google Soli [2] makes a huge break-

through pushing motion sensing to millimeter-level accuracy, enabling micro-gesture recognition. This technology enables finger control application in portable and wearable devices. Soli requires a dedicated RADAR-like chipset that needs to be integrated with the existing hardware. This motivated us to explore the following question: Can we detect fine gestures with conventional wireless signals, e.g. WiFi? If this is possible then we won’t need separate hardware for communication from gesture recognition. Shown in Fig. 1, we expect users to control their portable device with finger motion nearby. We believe such a technology could enable a set of new applications [15].

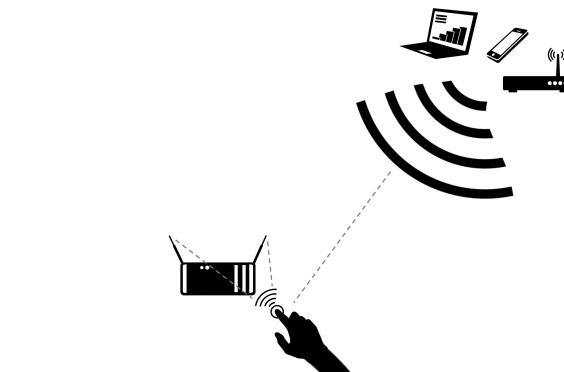


Figure 1: The system scenario of Mudra.

through pushing motion sensing to millimeter-level accuracy, enabling micro-gesture recognition. This technology enables finger control application in portable and wearable devices. Soli requires a dedicated RADAR-like chipset that needs to be integrated with the existing hardware. This motivated us to explore the following question: Can we detect fine gestures with conventional wireless signals, e.g. WiFi? If this is possible then we won’t need separate hardware for communication from gesture recognition. Shown in Fig. 1, we expect users to control their portable device with finger motion nearby. We believe such a technology could enable a set of new applications [15].

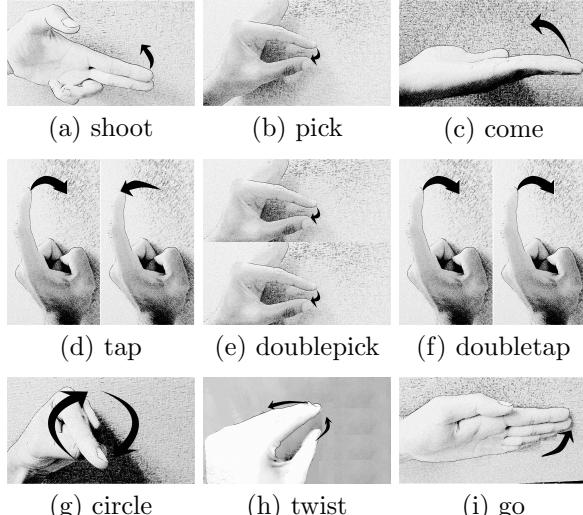
One obvious application is the interaction with wearables that have limited screen space. With nine gestures, users can easily scroll through one page, click button and switch among applications. Let’s imagine a reader is reading an e-book on apple watch. He can move to next page with a ‘tap’ and return back with ‘double tap,’ circling his finger to get to the bottom. It can also enable complex and in-the-air interactive games on portable devices. Users can control the action of figures with finger gestures in the air, e.g. going ahead, returning back, jumping and punching. Mobile devices increasingly appear in assisting health workers with ongoing medical education. While handling infectious medical task, e.g. blood test, touching their device would incur contamination. Now, we can enable

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*CoNEXT ’16, December 12-15, 2016, Irvine, CA, USA*

© 2016 ACM. ISBN 978-1-4503-4292-6/16/12...\$15.00

DOI: <http://dx.doi.org/10.1145/2999572.2999582>



**Figure 2:** Gestures considered in Mudra.

gesture recognition on the device using wireless signal eliminating the need for direct contact.

In this paper, we develop a novel finger-gesture recognition system, Mudra utilizing WiFi signals. Mudra just needs regular WiFi data transmissions either from local or remote sources. In this way, Mudra doesn't need to generate special signals (like Soli), thus, saving power for power-hungry devices [1, 19] and not hurting communication opportunities. Service of Mudra is always available as long as there are active transmissions in vicinity: Mudra can use a nearby WiFi-enabled desktop or laptop's transmissions to its access point (AP) to detect gestures. Furthermore, Mudra can use transmissions from multiple sources over time.

A key feature of Mudra is its training-free nature. This makes our system more promising in reality compared with other micro-motion technologies [9, 14, 23–25], all of which require training for specific location and user. One advantage is Mudra is not constrained by stationary scenario. That means, even if the user moves to another place, Mudra is still able to provide gesture recognition service using a WiFi transmitter at a new location without reconfiguration. Two key technologies to enable this feature are signal cancellation and gesture design and recognition principle. Signal cancellation renders us a motion indication which is highly sensitive to tiny movement and linearly relative to signal path delay. These two features together support us to apply distinct waveform shape of indication time series as recognition principle. Due to the relationship between delay and indication, those shapes could be consistent with different surrounding environments.

We implement Mudra with two receive antennas capturing signals from a COTS WiFi device. With 802.11n protocol [28], MIMO devices with multiple antennas are very common. Thus, Mudra can be implemented on COTS receiver if manufacturers open access to the samples. Mudra tracks finger-motion by deriving motion

indication with cancellation between two received signals. To make Mudra training-free, we select a set of gestures (Fig. 2) while more gestures can be added in the future. Besides, waveform consistency studies (Sec. 3.4) let us know how we can maintain waveform shapes across different users and locations. After that, Stretch Limited DTW algorithm is proposed to classify gestures with variation in duration and shape.

There are the following **challenges** in building Mudra:

- Finger-gestures are hard to detect because they result in subtle changes in the wireless channel.
- To utilize regular WiFi signals, we confront with lots of critical issues. Firstly, WiFi packets are typically not continuous. We need to implement smart packet detect/capture module in Mudra. Second, short subcarrier length and variable packet lengths in WiFi limit the tracking precision. Besides, 20MHz sampling rate poses a great computation challenge on packet alignment from two receiving antennas and real-time processing on host.
- User's movement changes the channel between the source and receiver. This makes finger-gesture recognition really hard and training for every location is not favorable for a user-friendly system.

**Contributions** Our contributions in this paper are as follows:

- We take the first step to look into finger-gesture recognition using conventional ISM band signals (from WiFi).
- Our system is able to utilize regular WiFi signals. To achieve this, our system needs to solve the above-mentioned challenges.
- Our system accommodates various scenarios and eliminates the need for training. Fig. 2 shows the set of gestures selected in Mudra that allows our system to be user-independent and location-independent.
- We build a prototype on NI based SDR platform. Our evaluation uses signals from COTS WiFi sources. We evaluate our system in various environmental conditions and system settings across multiple users. It shows that our system can achieve an average accuracy of 98% with a local source (2cm from Mudra) and 96% with remote source(s) (0.5–7m from Mudra).

**Features:** Mudra has the following features:

- It doesn't require training per user or per environment (Sec. 3.4).
- It supports both local and remote WiFi sources for gesture recognition (Sec. 3.3.2).

- It just needs conservative WiFi transmissions to perform very well (Sec. 6.1.3).
- It can combine signals from multiple WiFi sources and improve accuracy (Sec. 4.2.3).
- Multiple Mudra systems can operate simultaneously without affecting each other (Sec. 6.2.2).

## 2. RELATED WORK

Human tracking and motion detection have been broadly studied in the literature. We discuss existing techniques and their shortcomings in our setting.

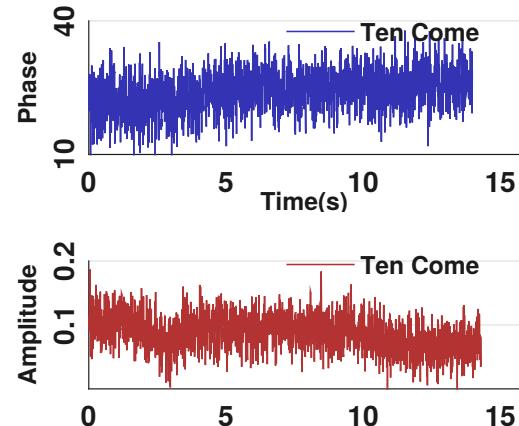
**Non-RF based micro-gesture recognition:** Imaging-based systems (e.g. Xbox Kinect [4], leap motion [3] and Maestro [10]) use monochromatic infrared cameras and LEDs to build body-depth imaging. Both Radar and infrared systems require line-of-sight (LOS) operation. Sonar-based system [18] requires specially modulated sound wave and exclusive devices. Mudra, on the other hand, utilizes regular WiFi signals in the ISM band, operating in parallel with WiFi communication. Furthermore, Mudra supports both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios.

**RF based micro-motion tracking:** Another set of works exploit RF signals to track human’s minor movements. Radar based systems (e.g. Google Soli [2]) show the ability to track minor finger movement by constructing Doppler profile using 60GHz radar signals. Such systems, however, require embedded chip to generate/capture and process radar signals. Besides, it is limited to line-of-sight (LOS) scenario due to the extra-directional feature of radar signals.

[9, 14, 23, 25] look into the keystroke detection problem. Mudra differs from them fundamentally in that Mudra is targeted at identifying finger gestures, not keystroke action. More importantly, Mudra is a user-friendly system without the need for training. Whereas, the others require location-specific or user-specific training for classification. One recent work detects minor keystroke motion by canceling two signals from two receiving antennas [9]. This system can not be applied to WiFi signals since (i) it assumed continuous signal transmissions and (ii) assumed the frequency response to be flat across the whole band. Therefore, they generate continuous wireless signal from SDR platform and only estimate the overall phase and amplitude of the channel. WiFi transmissions are not continuous and do not have a flat frequency response due to a small number of subcarriers spanning the whole band. The second issue affects cancellation quality, which is what Mudra relies on to identify finger-gestures. Mudra enables micro-gesture recognition with WiFi signals after solving those problems by implementing smart packet capture and estimating full channel state information (CSI). Mudra also resolves additional issues for WiFi signals such as variable length of packets.

[23, 25] make use of sound and electromagnetic signals emanating from keyboards, a feature unavailable

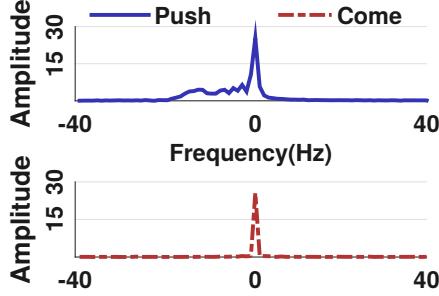
with finger motion. [14] mainly use Principle Component Analysis (PCA) to remove noise based on correlated variation of CSI with different keystrokes. It is highly unlikely that such correlation still works with various finger gestures since finger movements cause subtle changes to CSI: Fig. 3 shows the amplitude and phase variation over time when the user performed the same gesture 10 times. The plot shows no periodic variations. Another line of approaches (e.g. [24]) use MIMO beamforming, using multiple antennas and step motors to project signal to target. Target locating needs 6s and 85% of accuracy greatly affects overall accuracy. Mudra, on the other hand, doesn’t require such a luxurious infrastructure.



**Figure 3: Phase/Amplitude variation in channel.** When fingers repeat the ‘Come’ gesture ten times, the corresponding CSI amplitude/phase change is buried within large noise, making it infeasible for finger-gesture recognition. This is the same with other gestures while we just show ‘Come’ to save space, same with Fig. 4.

**Coarse-grained gesture/motion detection:** A large body of works track and identify larger-scale gesture motion (e.g. human limbs), which typically spans over range of several decimeters or target at the whole body, using characteristics of wireless signal. [20, 22] enables gesture recognition using Doppler profile generated from multi-second FFT to get sub-Hertz granularity. However, typical finger gestures are much shorter (e.g. tap just spans 0.5s), making this approach infeasible. To see this, we implement Wisee [20] on PXIe-1082 platform and see no observable Doppler shift with finger motion in Fig. 4. Frequency modulation was used with sound wave in [17] and carrier wave in [7] to track human breath. Mudra differs in that we don’t change the regular source to generate a delicately-designed signal, which can only serve recognition purpose.

Channel CSI / RSSI-based systems [15, 26] use coarse CSI feature to extract motion information, which is not feasible in detecting slight finger motion (Fig. 3). As to antenna array based technologies (e.g. [6] and [13]), WiDeo [13] spans four-antenna infrastructure covering



**Figure 4: Doppler profile in frequency domain.** Large gestures such as Limb Push produce noticeable Doppler shift (the above one) while figure gesture Come can not induce discernible shift, which is the same as no motion.

a length of 18cm, which is not available in portable devices. Wi-Vi [6] assumes a constant motion speed (1m/s) simply to identify whether the object is walking towards or away from a receiver. Another line of works get high positioning accuracy by attaching object with specific devices([16, 29, 30]). Tagoram [29] locates at a centimeter level using phase information of RFID backscatter signal. [30] tracks smartphone as a mouse by active sound wave transmission. Mudra is a user-friendly system which doesn't require such intrusive body instrumentation.

### 3. SYSTEM SETUP AND PRINCIPLE

To understand how exactly Mudra works, we will discuss three main issues in this section. First, we introduce the system scenario and setup. Next, we present the principle of motion tracking and then address various issues. Lastly, we explore consistency of waveform to understand how Mudra recognizes gestures without training.

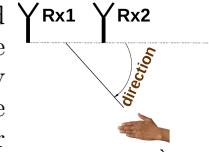
#### 3.1 System Scenario and Setup

With Mudra, we are designing a system that enables micro-gesture recognition in vicinity. As Fig. 1 shows, the hand moves roughly 7cm to the receiving antennas over 2-4cm with gestures in Fig. 2. This distance is measured as the minimum distance between the hand and the two-antenna segment. On the target device, there are two antennas capturing incoming WiFi signals. As to the distance between them, it is a critical design choice and will be studied in Sec. 6.1.2, which gives 10cm as a default setting for the best sensitivity while noting that Mudra can also work with shorter distance (Sec. 7).

Mudra can work with a remote source such as desktop, tablet and AP, and also a local source, i.e. an extra transmit antenna on the target device itself. Thus, distance between signal source and Mudra receiver could be ranging from several centimeters to a dozen meters. The motivation behind utilizing existing signals is two

folds. On the one hand, it eliminates the need for special transmissions for gesture recognition. On the other hand, including distant sources allows Mudra to utilize different sources over time and provide gesture recognition service throughout.

Two directions relate to finger motion in this paper. One is the hand direction which is measured as the angle relative to the line formed by two antennas, increasing clockwise (Fig. 5). The other is the finger moving direction. User experiences show that finger moving along forearm with gestures in Fig. 2 is comfortable and adopted naturally. Note that “moving along forearm” is a 2-D description as the ‘shoot’ rotates the wrist while making the fingers move along forearm in horizontal plane. Thus, we measure the forearm direction relative to the two-antenna-line as moving direction while telling the users to move along forearm. In Fig. 5, moving direction is 0 degree with forearm being parallel with the two-antenna-line.



**Figure 5:** Direction Demonstration

#### 3.2 Principle of Motion Tracking

In order to track minor finger motion, what indicator does Mudra use that is extra-sensitive to finger position?

##### 3.2.1 Channel Estimation and Signal Equalization

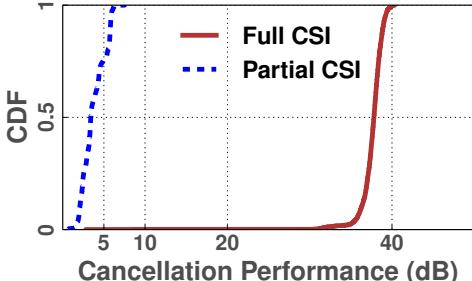
Mudra is equipped with two receiving antennas as shown in Fig. 1. Each of the antennas captures incoming signal samples, denoted as  $s_1(t)$  and  $s_2(t)$  respectively. The first thing is to equalize  $s_1(t)$  and  $s_2(t)$  for cancellation purpose. The solution here is to estimate the relative channel response between them.

To explain this, let two channel frequency responses be  $H_1(w)$  (' $w$ ' is angular frequency) and  $H_2(w)$  respectively, transmitted signal be  $T(w)$  and two received signals be  $S_1(w)$  and  $S_2(w)$ . Then,  $S_1(w) = T(w) * H_1(w) = T(w) * H_2(w) * \frac{H_1(w)}{H_2(w)} = S_2(w) * \frac{H_1(w)}{H_2(w)}$ . As it shows, signal  $S_2$  is equalized to  $S_1$  by compensating for the relative channel response  $\frac{H_1(w)}{H_2(w)}$ . We denote the time domain signal as  $\bar{s}_2(t)$  after this equalization. In Mudra,  $\frac{H_1(w)}{H_2(w)}$  is transformed to time domain impulse response then put into FIR filter as coefficients to equalize  $S_2$ .

In Mudra, since WiFi signals exhibit non-flat relative channel response for two receivers, we need to estimate  $\frac{H_2(w)}{H_1(w)}$  across the whole band. This is totally different from [9] which estimates just amplitude/phase difference with the flat-response assumption. In Fig. 6, the result validates the effectiveness and necessity of full channel estimation.

##### 3.2.2 Motion Indication with Signal Cancellation

After we obtain  $\bar{s}_2(t)$  from equalization,  $s_1(t)$  can be canceled from  $\bar{s}_2(t)$ . However, we want an indication



**Figure 6: Cancellation Performance Comparison.** We test in various scenarios. ‘Full CSI’ estimates on the whole band while ‘Partial CSI’ just calculates overall amplitude/phase difference.

which changes with finger moving. To this end, inspired by previous work [9], manual delay  $\Delta t$  and manual phase  $\phi$  are introduced to  $\bar{s}_2(t)$ . Now, when finger moves, an extra delay  $\delta\tau$  is introduced (Eq. 1a). Then, their frequency domain components ( $S_1(w)$  and  $\bar{S}_2(w)$ ) are related as shown in Eq. 1b.

$$s_1(t) = \bar{s}_2(t - \Delta t - \delta\tau)e^{-j\phi} \quad (1a)$$

$$S_1(w) = \bar{S}_2(w)e^{-jw(\Delta t + \delta\tau)}e^{-j\phi} \quad (1b)$$

Thus, as  $s_1$  is subtracted from  $\bar{s}_2$ , the cancellation is perfect (i.e. zero value) only at specific frequency points where the phase shift is an integer multiple of  $2\pi$ .

$$w^* = \frac{2\pi k - \phi}{\Delta t + \delta\tau}, k \in \mathbb{Z} \quad (2)$$

$w^*$  is a set of frequencies where the cancellation will be perfect.

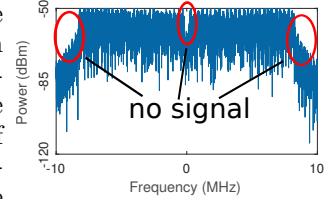
Since at frequencies farther away from  $w^*$  cancellation becomes worse (due to larger phase mismatch), we would get a trough at  $w^*$  shown in Fig. 8(a). Now, as the finger moves,  $\delta\tau$  varies accordingly causing trough location to change. Taking ‘Come’ (Fig. 2(c)) for instance, when fingers move from Rx1 to Rx2 and then back from Rx2 to Rx1,  $\delta\tau$  decreases first then increases to origin, which translates to increasing and decreasing of  $w^*$ . Then, we get waveform as shown in Fig. 10(c).

One could expect that a sharp trough gives us precise tracking, enabling gesture recognition with good performance. This relies both on: *perfect cancellation performance* (trough depth), which is guaranteed by our full CSI estimation, and *high sensitivity* of cancellation performance on phase mismatch (trough slope). We refer readers to [9] which has a study on sensitivity of cancellation degradation with phase mismatch.

#### How to select manual delay $\Delta t$ and manual phase $\phi$ ? - based on WiFi frequency structure.

In the OFDM physical layer of 802.11 protocol, there are 52 subcarriers over 20MHz band and no data at the center and edges (Fig. 7). Thus, if trough moves through these non-signal frequencies, we lose the track of finger motion. Mudra deals with this problem in two steps.

First, from Eq. 2, we can see the trough varying rate with motion drops with larger  $\Delta t$ . Thus, by setting a large enough  $\Delta t$ , we limit the varying range of trough. Second, we naturally want to make use of the maximum continuous Power (dBm) left or right half band. Thus, manual phase  $\phi$  is chosen as to tune the initial trough location to the center of left or right half band.



**Figure 7:** WiFi spectrum left or right half band. Thus, manual phase  $\phi$  is chosen as to tune the initial trough location to the center of left or right half band.

### 3.3 Various Challenges in Practice

#### 3.3.1 Realistic Issues with WiFi signals

##### 1. Fewer subcarriers in WiFi and cyclic prefix (CP)

In 802.11 a/g/n protocol, an OFDM symbol has just 64 subcarriers in 20/40MHz band. This is a critical issue which is not discussed in [9] as they send special OFDM symbols with 8192 subcarriers continuously. If we just use 64 subcarriers, then the frequency resolution is too coarse to perform gesture recognition. Here, we note that WiFi signals have power over all frequency points in its band because OFDM symbols are different from each other (shown in Fig. 7). So we are safe to do a large size FFT by simply connecting multiple OFDM symbols. However, do we need to avoid CP when combining multiple symbols together? The answer is no. In Mudra, signal copies on the receive antennas are for cancellation, not decoding. Thus, CP would not affect cancellation result as it also complies to channel response. Compared with WiSee [20], which equalized and combined OFDM symbols to enable fine frequency granularity, Mudra eliminates equalization because it relies on heterogeneity among those symbols.

##### 2. Various packet lengths in WiFi signals

Packet length decides how large the FFT could be: Larger this window better the frequency granularity. WiFi transmissions could have variable packet lengths causing the granularity to change from packet to packet. In Mudra, we utilize two signal processing properties for this issue, Zero-fill Invariance and Connect Invariance.

##### • Zero-filling Invariance

When packet length is a little less than the expected FFT length, we want to maintain the same granularity. The approach here is to use zero-filling at the end. In Eq. 3, we show the zero-filling invariance property, which means zero-filling doesn’t change the value on the same frequency. **DTFT**:

$$S(w) = \sum_{n=0}^{N-1} s[n]e^{-jwn} \quad (3)$$

This is Discrete Time Fourier Transform (DTFT).  $s[n]$  can be seen as the original signal samples with any length of zero-filling at the end.  $S(w)$  doesn’t change with those filled zeros.

### • Connect Invariance

Zero-filling just increases samples in frequency domain while not really contributing to signal spectrum. That means, although we get more values in frequency domain with zero-filling, the precision doesn't change because the trough is smoothed at the bottom. Thus, we have to drop those packets which may frequently appear in transmission, greatly sacrificing trough tracking opportunities.

However, we note the connect invariance property, which means trough location doesn't change if we combine residual signals of two neighboring packets. In Eq. 4,  $r1, r2$  are cancellation residuals from adjacent packets, which are connected to get  $r[n]$ . **DTFT:**

$$\begin{aligned} R(w) &= \sum_{n=0}^{N+M-1} r[n]e^{-jwn} \\ &= \sum_{n=0}^{M-1} r1[n]e^{-jwn} + \sum_{n=0}^{N-1} r2[n]e^{-jw(n+M)} \quad (4) \\ &= R1(w) + e^{-jwM}R2(w) \end{aligned}$$

If  $R1(w)$  and  $R2(w)$  have trough at the same frequency, then  $R(w)$  will also have trough at that frequency. Thus, by connecting neighboring packets after cancellation, we get the desired precision. Since motion is continuous in space, troughs within adjacent packets would share adjacent positions.

### 3. Noncontinuous transmissions of WiFi signals

Transmissions in WiFi are not continuous. For this, we implement a smart packet capture module with power threshold. Inter-packet interval would affect timing resolution of motion indications. When packet's interval is smaller, we get finer motion indications and thus, the recognition performance would be better. Fig. 18 shows experimental results on the effect of inter-packet interval on recognition performance.

#### 3.3.2 Frequency/timing Offset, Multi-path Effect and Irritable Neighbor as well as Local Source

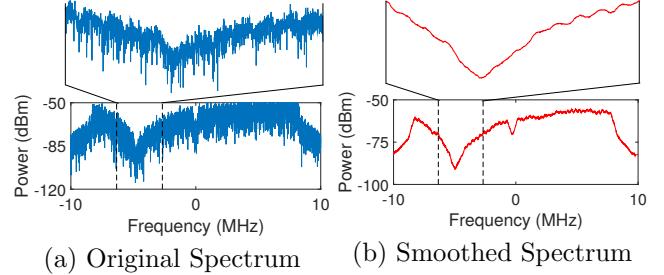
##### 1. How does Mudra deal with frequency/timing offset with remote source?

With remote source, frequency and timing offsets between transmitter and receiver ([8, 12, 21]) would impose a varying phase shift in the received signals, which can disable our system. In Mudra, however, two receiving antennas are synchronized with the same RF/sampling clock and timing offset as they are on the same device. Since our scheme obtains motion indication from cancellation of these two, we then get out of the phase shift problem.

##### 2. Multi-path effect in received signals

In experiment, we find that multi-path effect would incur signal destruction at some frequency, causing a trough similar to the one from cancellation (shown in Fig. 8(a)). This trough will remain in the cancellation result, thus, confounding with the desired one. In

Mudra, adaptive strategy is proposed to move desired trough out of confounding range. Specifically, we can detect multi-path trough from original received signals, the one without manual delay/phase. If it is in the left half band, we let the desired trough from cancellation be in the right half band by adjusting the manual phase (described earlier).



(a) Original Spectrum

(b) Smoothed Spectrum

**Figure 8:** Canceled signal spectrum before and after smoothing over 200 neighbors. The left plot is the power spectrum across 20MHz generated by original samples and the right one is a smoothed version by averaging over adjacent 200 values. The trough in each of them is magnified in a subplot on the top.

### 3. Irritable neighbor around trough

Fig. 8(a) shows that the spectrum after cancellation has irritable neighboring value around trough location, which are spikes in the magnified figure, seriously deteriorating the accuracy of trough location. We use neighboring average to remedy the problem and get a smoothed curve, shown in Fig. 8(b). The effectiveness and length choice of smoothing is discussed in the evaluation part, shown in Fig. 15(a).

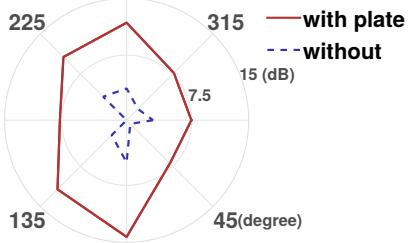
**Mudra with a local source** With a signal source on the same device, transmitting and receiving antennas are very close. Thus, the signal power over direct LOS path is much larger than reflection from hands, which means finger motion has a minor effect on received signals. Besides, due to near-field effect [27], the received signals see much noisier variation in trough recording.

**Solution of Mudra:** In Mudra, we deal with this problem by adding a metal plate as an electromagnetic partition between sender and receiver antennas so as to cut off the direct-path signals. Similar approach is adopted in [11] to implement absorptive shielding.

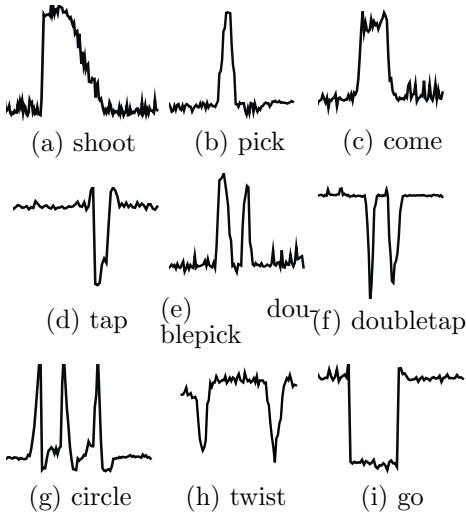
We test the effectiveness of this approach by comparing signal-to-noise ratio (SNR) in different directions with and without metal partition. We define SNR as the ratio between gesture-waveform and noisy variation amplitude. Fig. 9 validates that our method can increase SNR by 6 times on average. For fair comparison, finger moving direction is 0 for all positions.

### 3.4 Consistency Study of Gesture Waveforms

One of the goals of Mudra is to perform gesture recognition without any training and even when the user has moved to another environment. In this subsection, we



**Figure 9:** System SNR with and without iron plate. When signal source is very close (2cm), we test waveform-to-noise-amplitude ratio using ‘Come’ gesture in a circle in eight equi-angular directions.



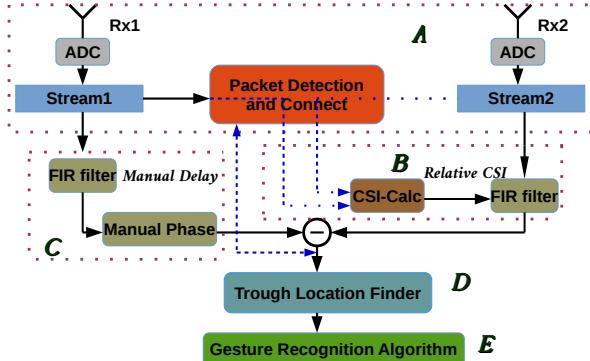
**Figure 10:** Waveform of gestures considered in Mudra.

study how the trough variation over time can be used to identify different gestures and study how sensitive this identification is to the environment.

Fig. 10 shows the trough location (in MHz) when plotted over time for different gestures (The axes are not shown due to space issues). Do these patterns remain the same across different environments? Do they change with different users?

When user moves, the environment around him would change. We care about different directions and distances of signal source in both LOS and nLOS scenarios. Thus, we choose eight equi-angular LOS positions with 2m distance and nLOS positions 5m away. For hand positions, we put hand at 0, 45 and 90 degree respectively. Note that we can ask the users to keep their hands at an angle that gives the best accuracy for all the finger-gestures. We use this best angle in our evaluations in Sec. 6.1.2.

Five users were asked to perform nine gestures at every source and hand position in a quiet office with no other people. We combine the results of all users in Table 1. **P1-P16** denotes sixteen distinct source positions with direction increasing starting from 0 degree: **P1-P8** are LOS while **P9-P16** are nLOS. ‘+’ means the waveform shape is same or similar with the same ges-



**Figure 11:** System model

ture in first position **P1** while ‘-’ represents direction flip of waveform shape. ‘\*’ is to say the corresponding waveform shape has no similarity with **P1**. In other words, ‘+’ or ‘-’ is good since the shape of the pattern is retained. However, ‘\*’ indicates that the pattern has changed and denotes a case against Mudra.

The results in Table 1 show that hand position in 90 degree could ensure consistency in gesture waveforms (Fig. 10) across all scenarios. While this is not surprising, as we know, there exists a linear relationship between signal phase and delay, and the reflection from fingers is a combination from all point objects. In later experiments, we will adopt this setting as default.

## 4. SYSTEM DESIGN AND GESTURE RECOGNITION

### 4.1 System Design

#### 4.1.1 System Model

Mudra has five main components (Fig. 11):

**A. Packet detect/connect:** This component captures packets from WiFi on two antennas. Since power level is not equal over samples, envelop of power trace is used for finding the beginning and end of continuous packet with a threshold. Mudra will discard packets shorter than 1000 samples to ensure a reasonable trough locating confidence. After that, neighboring packets will be connected together if packet length is less than 8192.

**B. Channel estimator/signal equilization:** As shown in Fig. 11 block **B**, channel estimator gets relative channel response between the two receiving antennas. Then, this coefficient is sent to an FIR filter, which equalizes the samples on the second antenna.

**C. Manual delay/phase injector:** As mentioned in Sec. 3.2.2 and 3.3.2, manual phase/delay is used to address WiFi frequency structure and multi-path issues. An FIR filter is used to inject manual delay to the captured packets followed by a manual phase addition.

**D. Trough location finder:** After cancellation, residual signal will be searched in the selected band and the minimum-value (frequency) position is the trough loca-

Position	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
0degree	+	*, -	+, *, -	+, *, -	*, -	+, *, -	+, *	+, *, -	+, *	+, -	+, *, -	+, *	+, *	*, -	+, -,	+, *, -
45degree	+	+, *, -	+, *	+, *, -	+, *	+, *, -	*, -	*, -	+, *, -	*, -	+, -,	+, *, -	+, *, -	+, *	+	*, -
90degree	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

**Table 1:** Consistency study of gesture waveforms with various signal source locations.

tion. Then, timing series of trough location would be sent to gesture detection/recognition module.

**E. Gesture recognition algorithm:** Gesture recognition module comprises of preprocessing, segmentation and classification. We implements online recognition with four gestures using simplified classification logic. To classify all nine gestures, we conduct gesture recognition off-line.

#### 4.1.2 Design detail

Here, we talk about how to choose design parameters.

##### 1. Length of FFT

Length of FFT decides the granularity we can look into in frequency domain. In 802.11 a/g/n protocol, the maximum size of PLCP Protocol Data Unit (PPDU) in physical layer is larger than 4000 bytes. Considering that one byte has 8 bits and 64-QAM modulation has 6 bits in one sample, we choose 8192 as the number of frequency points we look into, which is the length of FFT operation.

##### 2. Manual delay $\Delta\tau$ and manual phase $\phi$

As mentioned in Sec. 3.2.2, we tune the manual delay  $\Delta\tau$  to avoid trough running out of the selected range. After testing all target gestures by five testers, we choose 1, which means skipping one sample, as  $\Delta\tau$  to achieve largest sensitivity while satisfying this restriction, ensuring that it is feasible in almost all situations. After that, we move the trough location to the center of the range by choosing a value for  $\phi$ . Note that  $\phi$  is fixed after  $\Delta\tau$  is decided.

#### 4.1.3 Authorization Key for System Access

To avoid target device being miscontrolled by surrounding moving objects, we designed an authorization key containing multiple ‘Tap’ gestures as preamble for users to access this system. We have test in Sec. 6.2.1 to evaluate the effectiveness of different gesture repetitions.

## 4.2 Gesture Recognition

#### 4.2.1 Preprocessing

##### 1. Indication prediction with linear regression

The arrival of packets is not uniform over time, thus, causing irregular sampling of the waveform. We use linear regression to estimate the missing piece of the waveform with timestamp indicating arrival time of each packet. The predicting time points are based on 50 packets/second.

##### 2. Reduce noise variation

As wireless signals travel in the air, variations due to noise and environmental vibrations get into trough recording. We note that this variation is very random.

Thus, we use averaging in small neighboring range to reduce it without affecting gesture waveform. We average 10 neighboring values when capturing at 50 packets/second.

#### 4.2.2 Segmentation

In practice, we find that gesture waveform’s absolute value is not consistent over time because each motion doesn’t get fingers back to initial place exactly. Thus, Mudra uses derivation of indication time series for segmentation. The intuition of segmentation in Mudra is that, in derivation series, the gesture part usually holds a much larger value than non-gesture part. Thus, we use amplitude threshold to extract gesture waveform.

First, preliminary gesture waveform is located by using a conservative threshold  $T_1$  to get fragments belonging only to one gesture. Next, fragments in near scope will be bound together, forming a conservative range of one gesture. Then, we apply a smaller threshold  $T_2$  to obtain tighter gesture range with adjacent principle to existing one. Further, in order to ensure a complete waveform, we apply guard interval, which is 8 values at 50 packets/second, to expand in both sides.

#### 4.2.3 Classification

For gesture recognition, the remaining issue is how to match one segment with one gesture.

##### 1. Gesture Waveform Template

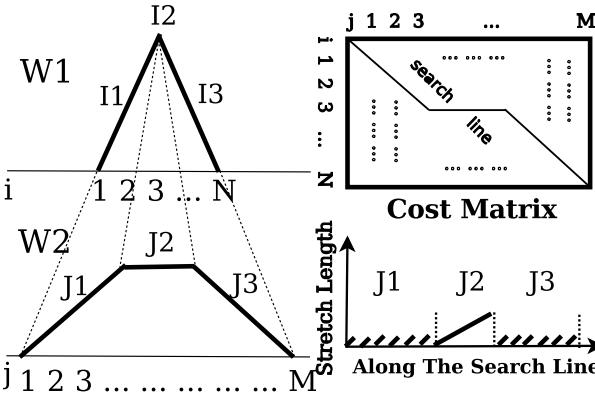
Mudra eliminates the burden of user training relying on distinct waveform shape of each gesture. Before testing, users were shown how to perform gestures. Each gesture in Fig. 10 has a typical duration, e.g. ‘Tap’ around 0.5s and ‘Come’ around 2s. Thus, we select a universal waveform template from piecewise linear function for each gesture independent with specific user.

##### 2. Stretch Limited DTW

Without a doubt, the waveform duration and shape of gesture would have variation across different users and multiple repetitions of one user. To solve this problem, we adopt Dynamic Time Wrapping (DTW) to get the minimum distance between segment and waveform template.

However, traditional DTW is not suitable. As DTW always tries to get minimum distance by unlimited stretching, some gestures such as ‘Come’ and ‘Tap’ would be mixed up with each other. To see this, let’s look at the left plots in Fig.12. By mapping parts I1, I2, I3 in waveform 1 (W1) to J1, J2, J3 in waveform 2 (W2) respectively, DTW gets a very small distance between W1 and W2 although they represent totally different shapes.

Here, we propose a novel algorithm - Stretch Limited DTW to solve this over-stretch issue. To be specific,



**Figure 12:** Stretch Limited DTW schematic diagram. The left plots show two waveform figures and how DTW map each part of them. The right plots show the walk through cost matrix and stretch length along the search line.

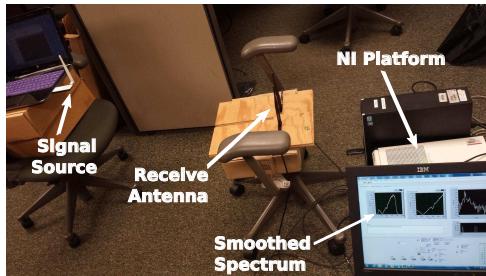
we introduce an additional cost proportional to stretch length. As to the algorithm, DTW first generates a cost matrix (shown in the right top of Fig.12) which contains the absolute difference between each pair of values in two arrays (which are  $W_1$  and  $W_2$  here) then walks through to find the path with least cost sum to the last. Here, we attach a penalty to the stretch through additional cost defined with stretch length (as shown in the right bottom of Fig.12). We set the scale factor to be the mean value of corresponding cost matrix.

### 3. Combine results from multiple sources

Our receiver system can rely on existing WiFi components on devices to fetch signal samples. In promiscuous mode, WiFi device can listen to packets from all sources. Thus, with packet decoding ability of WiFi adapter, our system could distinguish packets from different sources. To conceptually demonstrate how utilizing multiple sources could improve system performance, we simulate multiple traces using separately collected traces with sources in different locations. In classification, we use ground truth of waveform segments to include missing detection.

## 5. IMPLEMENTATION

We implement Mudra receiver system on the NI-based SDR platform. NI PXIe-1082 chassis is equipped with



**Figure 13:** Testbed

PXIe-8133 Express Controller and two NI-5791 FlexRIO adapters. Each adapter is connected with a VERT2450 3dBi gain antenna. In Virtex-5 based FPGA, DSP decimation after ADC with resolution of 14 bits generates 20MHz baseband samples. Direct Memory Access is built to transfer baseband samples from Rx1 to Rx2. Central controller is built on RTOS based PXIe-8133. We implement packet detect/connect and channel estimation on central controller.

In the FPGA of Rx2, we implement manual delay/phase injector on stream from Rx1, shown in system model Fig. 11. To equalize signal, samples from Rx2 are thrown into FIR filter with relative channel timing response as coefficients. On the central controller, two signal streams and cancellation stream are fetched from host FIFOs. Then, we use the captured packets to calculate relative channel response, sent to FIR filter in second stream pipeline with FPGA Module register map. Motion indication trace is generated online and stored to file along with timestamp indicating packet arrival time.

We design parallel architecture on central controller to solve computation challenge. First of all, we use multiple single-time **while** loops to distribute computation on different CPU cores. To avoid overflow of data transmission from FPGA to central controller, Mudra has a special loop assigned with highest priority dedicated for fetching data from host FIFO. Further, to maintain packet order, queue structure is used to pass data from one loop to next, forming producer-consumer structure.

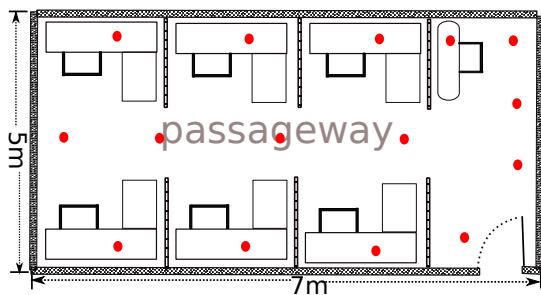
We use TP-LINK TL-WN722N adapter as signal source, with output power peak at 17.8 dBm [5], and NETGEAR WNDA3100 adapter as WiFi receiver. We build an adhoc connection between these two and enable UDP stream on Windows7 system. Fig. 13 shows our testbed.

## 6. EVALUATION

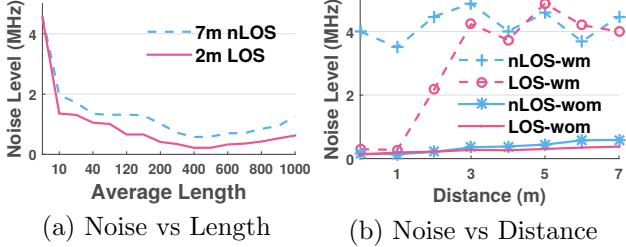
To study various factors, we conduct a set of experiments for each and make decision of system design based on the observation. Then, we evaluate our system holistically under different conditions.

### 6.1 Micro-Benchmarks

We set out to look into various performance-affecting factors. First, what is the noise variation level and how it varies with changing environment. Apart from this,



**Figure 14:** Floor plan



(a) Noise vs Length

(b) Noise vs Distance

**Figure 15:** Noise Variation: ‘wom’ is short for “without body movement” and ‘wm’ is short for “with body movement” between the source and receiver.

we also care about system sensitivity, distance between the source and the receiver and orientation. After that, impact of packet length/interval will be studied.

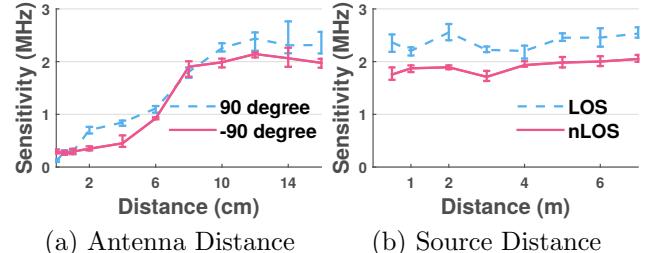
### 6.1.1 Noise variation

The irritable neighbor problem introduces severe noise in trough tracking, as mentioned in Sec. 3.3.2. To study this effect, we vary the distance between two antennas in Mudra. In a quiet office with door closed (floor plan shown in Fig. 14), we put source 2m away in LOS then 7m away in nLOS at 90 degree to receiver. For fair comparison, we just let one user to be around receiver statically. We measure indication varying range shown in Fig. 15(a). We observe that the average length (for smoothing) larger than 400 samples doesn’t reduce noise further. Thus, we select 400 as the default setting. We also find that noise gets even worse with more than 500. This is expected since averaging also makes the trough smoother.

Noise and environmental variation (e.g. human breath) would also induce noise variation. For this test, we choose different distances between signal source and receiver. To study the effect of moving body, another tester will move 3m away from receiver, blocking direct-path signals at some points. As indicated in Fig. 15(b), the noise increases with distance with no moving body. While, with moving body, noise gets extremely serious, disabling our system: We show how such scenarios can be detected and the corresponding samples can be avoided while performing gesture recognition in Mudra later. Another observation is that, with direct-path signal, moving body keeping 1m away from LOS generates a tolerable noise, while nLOS scenario doesn’t share this feature.

### 6.1.2 Sensitivity

We evaluate sensitivity of our system using WiFi signal in different scenarios. First, we study the factor of distance, including antenna distance and source distance. Then, we explore the impact of orientation i.e. hand/source direction and finger moving direction. We ask the users to perform ‘Come’ with 6 rounds. For each round, we obtain sensitivity as the average amplitude of (trough location) waveforms.



(a) Antenna Distance

(b) Source Distance

**Figure 16:** Sensitivity Study. **Takeaway:** Sensitivity of Mudra changes with antenna distance while keeping stable with source distance.

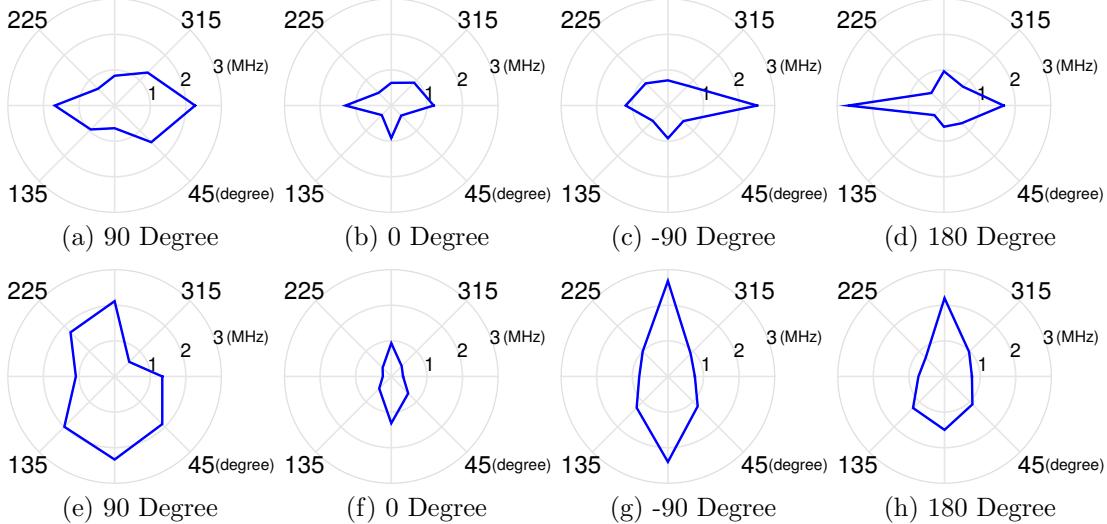
### (a) Versus distance

To study antenna distance, we put the source 2m away at 90 degree and -90 degree in LOS. The user is asked to perform ‘Come’ moving in 0 degree i.e. along two antennas with left hand position of 90 degree. Then, we gradually change distance between two antennas for each test. Fig. 16(a) indicates that sensitivity increases with antenna distance and stops at about 10cm. Considering that 10cm is a favorable scale for most portable devices, we select 10cm as the default setting throughout the paper.

After that, we continue to look into source distance. This time, we let the source be at 90 degree. We adjust the distance between source and receiver both in LOS and nLOS. To fairly compare sensitivity, we let user vary hand position and moving direction in eight equi-angular orientations, choosing the largest one. Fig. 16(b) shows that sensitivity with LOS is greater than nLOS. To our surprise, sensitivity almost doesn’t change with distance in both scenarios. We believe the reason is that sensitivity is mainly affected by distance between hand and receiver. Those results, combined with study on noise variation (Sec. 6.1.1), also validates the feasibility of finger-gesture recognition using WiFi signals.

### (b) Versus orientation

We first study the effect of finger moving direction. To this end, the user moves his fingers in eight equi-angular directions. We test with four equi-angular source positions with a distance of 2m away from receiver. For example, Fig. 17(a) represents a source at 90 degree. Hand is at 90 degree. As stated before, all directions are relative to two antennas in clockwise mode. In Fig. 17(a)-(d), we observe that moving in 0 degree guarantees at least 1MHz sensitivity for all source positions. Besides, moving in 180 degree provides comparable overall sensitivity as in 0 degree. This is expected since both directions are along two antennas. Then, we seek to evaluate the effect of hand position. We put hand in eight equi-angular positions. This time, however, the user tries to move in all directions and selects the largest one as sensitivity measurement. For example, in Fig. 17(e), with hand position at 90 degree, moving in 0 degree generates largest waveform amplitude. Fig. 17(e)-(h) show that putting hand vertical to



**Figure 17: Orientation Effect.** Upper four plots study moving direction; bottom ones study hand position.

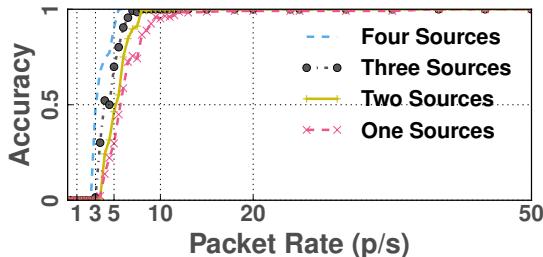
antenna, in all eight positions, has the best overall sensitivity. We have roughly the same observations across other gestures, thus, pushing us to select 90 degree for hand position and 0 degree for moving direction, which means, fingers move along two antennas with hand at vertical position. With such a default setting, combining with waveform consistency study in Sec. 3.4, we desire to make our system feasible with best recognition performance.

### 6.1.3 Packet length/interval

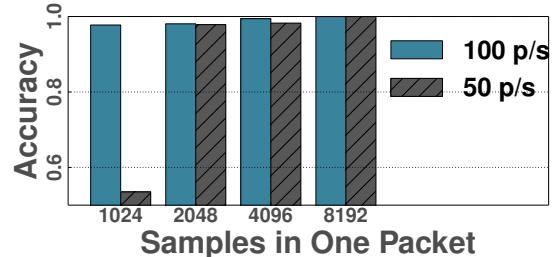
We run experiments to understand the effects of these parameters on recognition performance. Specifically, we setup an ad hoc connection between two wireless adapters and enable UDP streaming on sender side. With socket programming, we can adjust the length of character string which is sent as a bunch. On both the SDR platform and Wireshark packet capturer, we validate the packet size. Inspired by this, we adjusted packet rate by tuning the while loop interval. Unfortunately, the maximum frequency we get is around 100 packets/second. If we set a lower delay, packet rate increased drastically in a burst mode, which means large portion of duration has no transmission. While, a good thing is that our system already has favorable classification accuracy with 50 packets/second. Thus, such a limitation doesn't affect our evaluation. Signal source is put 2m away at 90 degree position. We have five

users perform four gestures - ‘Come, Go, Tap, Pick,’ each with 50 repetitions. We implement a simplified algorithm on SDR platform to recognize these gestures in real time. Specifically, a threshold is applied to derivation of the indication trace after reducing noise, where boundaries of gesture is detected. Then, we get gesture duration using the attached timestamp. Since ‘Come, Go’ last much longer than ‘Tap’ and ‘Pick’ (2s vs 0.5s), we can distinguish between them. Further, by utilizing the waveform direction, we come to the final decision by judging whether the first derivation is positive or negative.

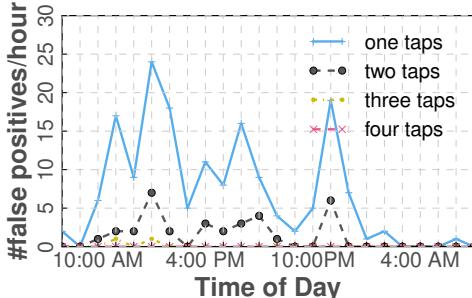
To evaluate interval effect, we transmit packets at different rates. Fig. 18 shows how classification accuracy changes with packet rate. Let's first look at the resulting line with just one signal source. Here, accuracy is as high as 0.95 with just 10 packets/second - typical WiFi beacon rate, occupying 0.4% of the whole duration. This result means that our system just needs very conservative WiFi signals. We note that no gesture can be detected with rate less than 3 packets/second. This is because such low rate would miss gesture waveform. Then, we apply multiple signal sources and combine results. There are four sources located in four directions. The results indicate that with more signal sources we can get better performance. One reason is that with more sources we reduce the rate of missing detection.



**Figure 18:** Impact of packet interval.



**Figure 19:** Impact of packet length.



**Figure 20:** False detection rate during a whole day.

Another reason is that with majority voting, large errors with specific source could be avoided. Typically, with four sources, accuracy could be 0.95 same as one source while with just 5 packets/second per source.

As to the packet length, we evaluate recognition performance with four different choices. Packet rate is set to 100 packets/second and 50 packets/second. In Fig. 19, the results indicate that by connecting short packets, Mudra could be effective in gesture recognition. We note, compared with Fig. 18, that the performance is a little worse. This is because connecting packets not only reduces indication frequency but also degrades indication precision due to averaging over multiple packets. The drastic drop with 1024 at 50 p/s indicates the minimum indication length needed for good performance.

## 6.2 System Evaluation

### 6.2.1 False Positive

To avoid mis-detection with surrounding moving object, Mudra use multiple ‘Tap’ as authorization key. We evaluate its effectiveness in this section.

In our experiment, we put two antennas on one desk so that it can be affected by human regular activities. All members are free to do anything, walking, speaking or eating, in the office. We have four position settings which put source 5m away from receiver. Recording last 24 hours over a whole weekday from 8am to 8am, we get the result of false detection number per hour in Fig. 20.

The results show that when one ‘Tap’ is used as access key, the false detection rate over 24-hour period is 6.92 per hour. We note that this is very low. This is because ‘Tap’ lasts just about 0.5s, while typical human movements such as walking, eating and typing, last much longer. Besides, those large-scale motion usually push through out of legal range. Lastly, ‘Tap’ waveform is special which is not easy for random motion and environmental variation to generate. We also find that with multiple repetitive gestures as the key, the false rate decreases greatly. It is reasonable since we expect gestures in one key would be in a certain range. In this way, for two repetitive ‘Tap’ false rate is 1.37 per hour and using three ‘Tap’ makes it as low as 0.08.

### 6.2.2 Mudra Recognition Performance

**(a) Evaluation with Local Source:** Mudra can utilize signals from the target device with solution in Sec. 3.3.2. We put source (WiFi adapter) 2cm away from the first antenna along the line with iron plate to block the direct path. Receiver is put on desks with surrounding strong reflective objects and empty space near the door shown in Fig. 14. Five users, who don’t know how Mudra works, are shown how to perform each gesture. Each gesture in Fig. 2 is performed for 40 times by each user, following regulation of position and direction. During test, users sit in a chair. As to WiFi signals, we adjust packet rate to 50 per second with length of 4096, which means, indication after packet-connect would be 25 per second. We measure recognition accuracy combining detection and classification across all positions and users. Fig. 21 shows the average recognition accuracy is 98%. In this matrix, we observe that recognition has the largest error with ‘shoot.’ Checking the intermediate result, we find that waveform of ‘shoot’ is the most variable one across users.

	circle	come	doublepick	doubletap	go	pick	shoot	switch	tap
Actual Gesture Performed	1	0	0	0	0	0	0	0	0
circle	0.98	0	0	0	0.02	0	0	0	0
come	0	0.97	0	0	0.02	0	0	0	0.01
doublepick	0	0	0.97	0	0	0	0	0	0.01
doubletap	0	0	0	1	0	0	0	0	0
go	0	0	0	0	0.98	0.02	0	0	0
pick	0.01	0	0	0	0	0.99	0	0	0
shoot	0	0.02	0	0	0	0.03	0.95	0	0
switch	0	0	0	0.02	0	0	0	0.98	0
tap	0	0	0	0	0	0.01	0	0	0.98

**Figure 21:** Confusion Matrix with Local Source.

**(b) Evaluation with Remote Source:** To evaluate system performance with signals from remote device, we put transmitting WiFi adapter across positions in the office (dots in Fig. 14). With receiver on the walkway and on desk, we include scenarios with distances ranging from 0.5m to 7m both in LOS and nLOS. Each gesture is performed a total of 40 times. As Fig. 22 shows, the average accuracy is 96%. **Takeaway:** Mudra can deliver high accuracy with sources less than 7m away under quiet environment. The performance is worse than local source since larger distance can introduce more noisy variation. After that, we combine results of local and remote sources to look into performance with individual user (shown in Fig. 23): The accuracy is consistent across users showing that Mudra is generic and doesn’t need training.

#### • Hand-held Device and Multi-source Combining:

For portable devices, e.g. smartphone and tablet, people usually hold devices in hand when using them. It is a good utility if our system can support gesture recog-

	Actual Gesture Performed	circle	come	doublepick	doubletap	go	pick	shoot	switch	tap	
Gesture Classified	circle	0.97	0	0	0.02	0	0	0	0	0	0
circle	0.01	0.96	0.01	0	0	0.02	0	0	0	0	0
come	0.02	0	0.98	0	0	0	0	0	0	0	0
doublepick	0.02	0	0.98	0	0	0	0	0	0	0	0
doubletap	0	0	0	0.99	0	0	0	0	0	0	0
go	0.01	0	0	0.02	0.96	0	0	0	0	0.01	0
pick	0.02	0	0	0	0	0	0.98	0	0	0	0
shoot	0.01	0.03	0	0.01	0	0	0.94	0	0	0.01	0
switch	0.02	0	0	0.03	0	0	0	0	0.94	0.01	0
tap	0	0	0	0	0	0	0	0	0	0.98	0

Figure 22: Confusion Matrix with Remote Source.

nition in this scenario. For this exercise, we bundle two antennas with a stick at a distance of 10cm. Users were asked to hold the stick with right hand statically and perform gestures with left hand. To test in the worst case, we put source at distances of 6m and 7m in four nLOS locations. Observation from this test is larger noisy variations due to hand-holding. However, the result shows that average recognition accuracy reaches 90% since noise is still tolerable with users sitting in a chair.

Further, to explore how multiple sources could help in such scenario, we combine recognition with traces from those sources. As a result, the overall accuracy is improved to 96%. It validates the effectiveness of utilizing multiple sources.

#### • Operating with Other Users:

To test whether multiple users can perform gestures at the same time, we let one user operate on our system with other people nearby, also performing finger gestures. The interfering users are asked to be 2m away while facing to receiver. Also, we put the source across various locations. With three users nearby performing gestures, recognition accuracy is 94%. Thus, we are glad to show multiple users can operate simultaneously in the same room.

#### • System Adaptation:

By tracking noise level, Mudra can enable two functions and perform better. Firstly, noise level in trace from one source indicates the link quality between the source and receiver, thus, is useful for selecting the good ones. Second, by tracking noise level, system can judge random variation around so as to avoid using such observations. Note that combining observations from multiple sources comes in handy when one or some sources are affected by significant human movements. Here, we test effectiveness of the second one. We put source 5m away. Signal would be affected by human movements between the source and receiver. During working time, i.e. 9am - 5pm, users are asked to choose 10 time slots, only depending on the system notification, to do gestures. While others can normally act in the office, our system can achieve accuracy as 90%.

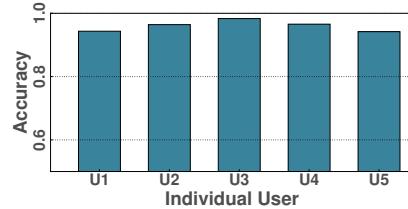


Figure 23: Recognition performance with individual user.

## 7. CONCLUSION AND DISCUSSION

In this paper, we take the first step towards designing a finger-gesture recognition system with WiFi signals and validate our design with COTS WiFi sources. Our system is user-friendly with no requirement of training. We believe such a technology would stimulate new applications, such as in-air interaction, untouchable-device control and disabled-friendly design. With both on-target local source and remote source, Mudra could support accuracy of at least 96%. While, our system could not support scenario with moving body nearby, we showed how such scenarios can be detected and avoided during gesture recognition.

#### (a) Mudra with closer antennas

There may be concern that it's hard to separate antennas on portable devices by 10 cm. Fortunately, sensitivity study in Sec. 6.1.2 reveals that shorter antenna distances still enable gesture recognition. To prove this, we test with 6 cm and 8 cm with source 2m away. In two cases, the accuracy is 86% and 93% respectively.

#### (b) Human interference

Surrounding human motion will incur great variation as shown in noise study Sec. 6.1.1. However, we also find that when body is far from direct path between source and receiver, noise level could be tolerant for gesture recognition. In personal spaces, such as office and living room, such requirements can be satisfied, i.e. people are separated by individual rooms with laptop/desktop nearby.

#### (c) Why don't we use multiple receivers?

Multiple-receiver design could solve the problem that sensitivity is low with specific position (as studied in [9]). While, in our problem, one critical issue is to ensure consistency of gesture waveforms. If we use multiple receivers, then the required position and direction couldn't be satisfied as in Sec. 3.4. However, single-receiver design doesn't suffer too much in our problem, i.e. sensitivity is favorable under various settings. We believe it is due to operating closer to receiver.

## 8. ACKNOWLEDGEMENTS

We thank our shepherd Ben Zhao and CoNEXT reviewers for their helpful feedback. This work was supported in part by the NSF grants CNS-1547306, CNS-1514260, CNS-1254032 and CNS-1302620.

## 9. REFERENCES

- [1] Findlay Shearer. Power management in mobile devices, chapter Hierarchical View of Energy Conservation, pages 32-75. Newnes, 2008.
- [2] Google. Soli. <https://www.youtube.com/watch?v=0QNiZfSsPc0>.
- [3] Leap motion. <https://www.leapmotion.com>.
- [4] Microsoft. X-box kinect. <http://www.xbox.com>.
- [5] TPLINK. TL-WN722N. [https://wikidevi.com/wiki/TP-LINK\\_TL-WN722N](https://wikidevi.com/wiki/TP-LINK_TL-WN722N).
- [6] ADIB, F., AND KATABI, D. Wi-Vi: See Through Walls with WiFi! In *ACM SIGCOMM*, 2013.
- [7] ADIB, F., MAO, H., KABELAC, Z., KATABI, D., AND MILLER, R. C. Smart Homes that Monitor Breathing and Heart Rate. In *ACM CHI*, 2015.
- [8] CHEN, B., QIAO, Y., ZHANG, O., AND SRINIVASAN, K. AirExpress: Enabling Seamless In-band Wireless Multi-hop Transmission. In *ACM MobiCom*, 2015.
- [9] CHEN, B., YENAMANDRA, V., AND SRINIVASAN, K. Tracking Keystrokes Using Wireless Signals. In *ACM MobiSys*, 2015.
- [10] DELL, N., D'SILVA, K., AND BORRIELLO, G. Mobile Touch-Free Interaction for Global Health. In *ACM HotMobile*, 2015.
- [11] EVERETT, E., SAHAI, A., AND SABHARWAL, A. Passive self-interference suppression for full-duplex infrastructure nodes. *IEEE Transactions on Wireless Communications* (2014).
- [12] GOLLAKOTA, S., AND KATABI, D. Zigzag decoding: combating hidden terminals in wireless networks. In *ACM SIGCOMM*, 2008.
- [13] JOSHI, K., BHARADIA, D., KOTARU, M., AND KATTI, S. Wideo: Fine-grained device-free motion tracing using RF backscatter. In *USENIX NSDI*, 2015.
- [14] KAMRAN ALI, ALEX X. LIU, W. W., AND SHAHZAD, M. Keystroke Recognition Using WiFi Signals. In *ACM MobiCom*, 2015.
- [15] KELLOGG, B., TALLA, V., AND GOLLAKOTA, S. Bringing gesture recognition to all devices. In *USENIX NSDI*, 2014.
- [16] KOTARU, M., JOSHI, K., BHARADIA, D., AND KATTI, S. Spoton: Indoor localization using commercial off-the-shelf WiFi nics. In *IPSN*, 2015.
- [17] NANDAKUMAR, R., GOLLAKOTA, S., AND WATSON, N. Contactless Sleep Apnea Detection on Smartphones. In *ACM MobiSys*, 2015.
- [18] NANDAKUMAR, R., IYER, V., TAN, D., AND GOLLAKOTA, S. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *ACM CHI*, 2016.
- [19] PERRUCCI, G. P., FITZEK, F. H., AND WIDMER, J. Survey on energy consumption entities on the smartphone platform. In *IEEE Vehicular Technology Conference*, 2011.
- [20] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-home gesture recognition using wireless signals. In *ACM MobiCom*, 2013.
- [21] RAHUL, H. S., KUMAR, S., AND KATABI, D. JMB: scaling wireless capacity with user demands. In *ACM SIGCOMM*, 2012.
- [22] TAN, B., WOODBRIDGE, K., AND CHETTY, K. A real-time high resolution passive WiFi Doppler-radar and its applications. In *International Radar Conference*, 2014.
- [23] VUAGNOUX, M., AND PASINI, S. Compromising Electromagnetic Emanations of Wired and Wireless Keyboards. In *USENIX SSYM*, 2009.
- [24] WANG, G., ZOU, Y., ZHOU, Z., WU, K., AND NI, L. M. We can hear you with Wi-Fi! In *ACM MobiCom*, 2014.
- [25] WANG, J., ZHAO, K., ZHANG, X., AND PENG, C. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *ACM MobiSys*, 2014.
- [26] WANG, Y., LIU, J., CHEN, Y., GRUTESER, M., YANG, J., AND LIU, H. E-eyes: device-free location-oriented activity identification using fine-grained WiFi signatures. In *ACM MobiCom*, 2014.
- [27] WIKIPEDIA. Near and far field — wikipedia, the free encyclopedia, 2015. [Online; accessed 20-October -2015].
- [28] XIAO, Y. IEEE 802.11 n: enhancements for higher throughput in wireless LANs. *IEEE Wireless Communications* (2005).
- [29] YANG, L., CHEN, Y., LI, X.-Y., XIAO, C., LI, M., AND LIU, Y. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *ACM MobiCom*, 2014.
- [30] YUN, S., CHEN, Y.-C., AND QIU, L. Turning a Mobile Device into a Mouse in the Air. In *ACM MobiSys*, 2015.