QGesture: Quantifying Gesture Distance and Direction with WiFi Signals

NAN YU, State Key Laboratory for Novel Software Technology, Nanjing University, China
WEI WANG, State Key Laboratory for Novel Software Technology, Nanjing University, China
ALEX X. LIU, Deptartment of Computer Science of Engineering, Michigan State University, USA
LINGTAO KONG, State Key Laboratory for Novel Software Technology, Nanjing University, China

Many HCI applications, such as volume adjustment in a gaming system, require quantitative gesture measurement for metrics such as movement distance and direction. In this paper, we propose QGesture, a gesture recognition system that uses CSI values provided by COTS WiFi devices to measure the movement distance and direction of human hands. To achieve high accuracy in measurements, we first use phase correction algorithm to remove the phase noise in CSI measurements. We then propose a robust estimation algorithm, called LEVD, to estimate and remove the impact of environmental dynamics. To separate gesture movements from daily activities, we design simple gestures with unique characteristics as preambles to determine the start of the gesture. Our experimental results show that QGesture achieves an average accuracy of 3 cm in the measurement of movement distance and more than 95% accuracy in the movement direction detection in the one-dimensional case. Furthermore, it achieves an average absolute direction error of 15 degrees and an average accuracy of 3.7 cm in the measurement of movement distance in the two-dimensional case.

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

Additional Key Words and Phrases: Gesture Recognition, WiFi Signals, Wireless Sensing

ACM Reference Format:
Nan Yu, Wei Wang, Alex X. Liu, and Lingtao Kong. 2018. **QGesture: Quantifying Gesture Distance and Direction with WiFi Signals.** ***Proc. ACM Hum.-Comput. Interact.*** **1, 4, Article 39 (March 2018), 22 pages. https://doi.org/0000001.0000001**

1 INTRODUCTION

Recently a number of interesting WiFi-based gesture recognition schemes have been proposed [1, 8, 19, 24, 29]. As human bodies are mostly made of water, they reflect WiFi signals and introduce distortions in the received signal when they move. Different gestures cause different types of distortions in WiFi signals. Thus, by analyzing the changes in WiFi signals, we can recognize the corresponding gesture. WiFi-based gesture recognition has many advantages over traditional approaches that use cameras [4] or wearable sensors [10, 28, 36]. For example, WiFi-based gesture recognition requires neither lighting nor carrying any devices. It also provides better coverage as WiFi signals can penetrate through walls.

Authors' addresses: Nan Yu, State Key Laboratory for Novel Software Technology, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China; Wei Wang, State Key Laboratory for Novel Software Technology, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China; Alex X. Liu, Deptartment of Computer Science of Engineering, Michigan State University, Computer Science and Engineering, USA; Lingtao Kong, State Key Laboratory for Novel Software Technology, Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu, China.

2573-0142/2018/3-ART39 $15.00
https://doi.org/0000001.0000001

One of the most important applications of WiFi-based gesture recognition is to interact with smart home devices. Existing home appliances use physical interfaces, such as knobs and levers, to provide quantitative inputs, including volume adjustment for TVs and brightness adjustment for lights. These physical inputs allow the user to fine-tune the input value based on immediate feedback. It is difficult to emulate these physical inputs using popular voice-based interactions provided by Amazon Echo or Google Home. However, WiFi-based gesture control can enable such fine-grained quantitative control. For example, the user can push his hand forward to increase the volume of the TV set, where the magnitude of volume increase is proportional to the distance of pushing. To enable this, we need not only to recognize different predefined gestures, but also to quantify gesture movement distance in a granularity of a few centimeters so that the system can adjust the volume according to the distance that the user pushes his hand, while providing audio feedback on the current volume setting along the pushing process. In this way, the user can quantitatively adjust the volume to the desired value using a single action rather than repeating the gesture to increase or decrease the volume by a small amount at each time.

The task of using Radio Frequency (RF) signal obtained from commercial hardware to measure the gesture movement distance and direction is difficult. Prior systems that use WiFi measurements from commercial devices often require whole-body movements, such as walking, to track movement speeds and directions [3, 24, 33, 37]. With the coarse measurement from commercial WiFi devices, existing schemes cannot trace hand/finger movements, which introduces weaker WiFi signal distortions than whole-body movements, in a fine-granularity.(delete?) They can only recognize hand/finger movements by matching them to predefined gesture patterns [1, 18]. Using customized ASIC chips based on the 60 GHz radar technology, Google's recent Soli system can quantify micro gestures so that those gestures can serve as human input for small wearable devices (such as smart watches) whose touch screens are too small for human to conveniently input [19]. However, due to the fast decay of 60 GHz signal in the air, 60 GHz system requires the gesture to be performed within tens of centimeters [31]. The limited operational range makes them unsuitable to serve as remote control interfaces for home appliances.

In this paper, we propose QGesture, a Quantifying Gesture distance and direction system, which uses Commercial-Off-The-Shelf (COTS) WiFi devices to measure the movement distance and direction of human hands. Figure 1 shows the basic system structure of QGesture. When the user pushes towards the target device, the device collects Channel State Information (CSI), which is perturbed by the WiFi signal reflected by the moving hand. The signal reflected by the hand appears as a dynamic vector component in the CSI values, which causes the complex-valued CSI measurement to rotate. The distance of movement can be calculated by the phase change of complex-valued CSI measurement and the direction of movement can be determined by the rotation direction. Therefore, the user can push forward to increase the volume while pulling away to reduce the volume and the amount of increase/decrease is determined by the movement distance. As the perturbation of WiFi signals can be captured at a long distance, QGesture can work at a distance as far as 2 meters. QGesture is the first step towards quantitative remote control for home appliances. It shows the feasibility of fine-grained distance/direction measurement of hand movement over a few meters using COTS WiFi devices. Note that currently only a limited modules of WiFi network cards can provide CSI measurements [12] and CSI is not available on smartphones. We envision that more commercial WiFi devices would open their CSI information so that our approach can be deployed on smartphones in the near future.

There are four key challenges that need to be addressed in designing QGesture.

• *Reconstruct the phase of CSI measurements:* The phase of CSI measurements is important for determining the movement directions [29]. However, due to hardware imperfections in COTS WiFi Network Interface Cards (NICs), there are Carrier Frequency Offsets (CFO) and Sampling Frequency Offsets (SFO) between the transmitter and the receiver [17, 34]. Both the CFO and SFO introduce high variations in the phase of CSI and these variations are sensitive to temperature and hardware conditions. Therefore, it is difficult to predict and remove such phase

variations without disturbing the small phase changes caused by hand movements. To address this challenge, we carefully analyze the phase offsets in different antenna pairs and design our phase correction algorithm so that phase changes of hand movements can be preserved. Hence, we can determine the movement direction with an accuracy of more than 95%.

• *Separate the channel state changes caused by the moving hands from the mixture of changes caused by other body parts:* This is particularly important for a gesture recognition system to operate over a long distance, *i.e.*, several meters, because such system captures both the gesture movements and the environmental dynamics. When the user performs the gesture, their torso and arms also move at the same time, which significantly perturb the measurements of the wireless channel. To address this challenge, we analyze the CSI signals and find the typical signal frequencies generated by gestures, which are different from those generated by movements of other body parts. We then design a robust estimation algorithm, called LEVD, to remove the impact of environmental dynamics.

• *Separate gesture movements from daily activities:* Daily activities, such as walking and sitting down, also distort the wireless channel state information. To ensure that QGesture only responses to the channel distortion caused by specific gestures, we design simple gestures with unique characteristics as preambles to determine the start of the gesture. Our experimental results show that QGesture can efficiently recognize the preamble with an accuracy of 92.5% and a low False Positive Rate (FPR) of 3.2%.

• *Accommodate arbitrary pushing angles:* The phase changes of CSI measurements are determined by the changes in path length, which depends on the movement angle and the position of the hand with respect to the sender and receiver. When the hand moves along the line connecting the sender and receiver, the path length changes by two times of the movement distance. However, when the movement is in other directions, we may get smaller path length change for the same movement distance. To allow pushing along arbitrary angle, we need to perform the 2D tracking of the hand. To address this challenge, we propose to use multiple receivers to track path length changes of different paths at the same time. By doing this, we can triangulate the position of the hand and measure both the pushing angle and the movement distance. Our experimental results show that we can measure the movement angle with an accuracy of 15 degrees and movement distance with an accuracy of 3.7 cm.

We implemented QGesture using COTS WiFi routers and laptops. Our experimental results show that QGesture can measure the gesture movement distance with an accuracy of 3 cm within a distance of 1 meters in normal indoor environments. QGesture can also reliably detect the hand movement direction with an accuracy of more than 95% in the one-dimensional case. Furthermore, it achieves an average absolute direction error of 15 degrees and an average accuracy of 3.7 cm in the measurement of movement distance in the two-dimensional case.

2 RELATED WORK

We classify existing related gesture systems into two groups: RF-based recognition/tracking and non-RF-based recognition/tracking. Considering the way of collecting RF signal, we further classify RF-based into two categories: RF-based recognition/tracking using COTS hardware and RF-based recognition/tracking using specialized devices.

RF-based Recognition/Tracking Using COTS Hardware: Most COTS hardware based on recognition and tracking systems uses the Received Signal Strength Indicator (RSSI) or CSI obtained from WiFi NICs to capture gesture signals [1, 5, 7, 13, 18, 22, 25, 26]. The WiKey scheme proposed to use CSI dynamics to recognize micro human activities such as keystrokes [6]. The WiFinger scheme used CSI to recognize a set of eight gestures with an accuracy of 93% [26]. The WiGest scheme used three wireless links to recognize a special set of gestures, where user hands blocked the signal and thus introduced significant RSSI changes, and achieved a recognition

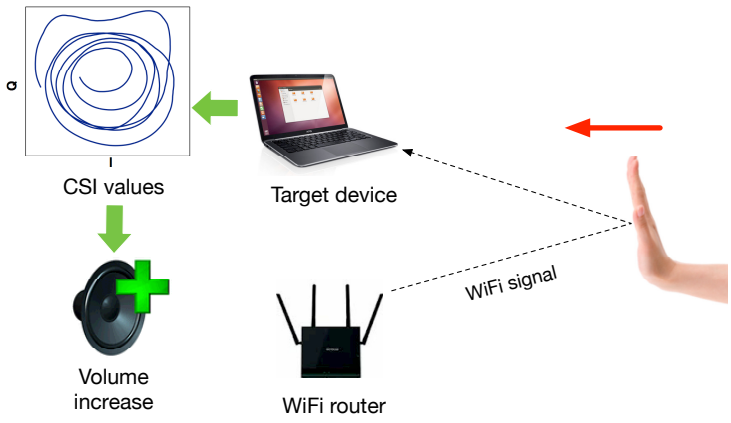


Fig. 1. QGesture system overview.

accuracy of 96% [1]. However, most of these systems only recognized a predefined set of gestures without considering movement distance/direction measurements. WiDir used WiFi CSI to estimate the whole-body movement direction, such as walking, with an error of 10 degrees [33]. For small hand movements, WiDraw used the Angle-Of-Arrival (AOA) measurement to achieve a tracking accuracy of 5 cm [25]. However, the AOA-based approach also had a limited working range of fewer than 2 feet, so that it cannot be used as remote controls in HCI applications. QGesture is inspired by previous WiFi CSI processing technologies, including the noise removal algorithm, basic phase correction algorithm, and preamble gesture design. QGesture advances the state-of-art design by capturing the small phase variations caused by hand movements at a long distance. In addition to WiFi-based schemes, existing schemes also use COTS RFID readers and tags to track gestures [9, 28]. However, these systems require users to wear RFIDs or operate close to the RFID array, which makes them inconvenient to use.

RF-based Recognition/Tracking Using Specialized Devices: RF signals can also be captured by specialized devices such as software radio systems. Software radio systems, such as USRP or WARP, have access to the fine-grained baseband signal so that they can provide the capability of quantifying hand/finger movement distance and speed [2, 3, 14, 31, 35]. WiSee used USRP software radio to identify and classify nine whole-body gestures with an accuracy of 94% [24]. WiTrack used specially designed Frequency-Modulated Continuous-Wave (FMCW) radar with a high bandwidth of 1.79 GHz to track human movements behind the wall with a resolution of about 11 cm to 20 cm [2, 3]. WiDeo used the WARP hardware to enable a tracking accuracy of 7 cm for multiple objects [14]. AllSee used a specially designed analog circuit to extract the envelopes of the received signals and recognize gestures within a short distance of 2.5 feet [15]. While these system provided valuable insights on the dynamics of the wireless signal, tracking with the coarse-grained CSI measurements requires a different set of signal processing algorithms.

Non-RF-based Recognition/Tracking: Gesture recognition can be enabled by non-RF based technologies, including computer vision, wearable devices, and sound waves. Computer vision based gesture recognition uses cameras and infrared sensors to reconstruct the depth information from videos. The distance measurement accuracy for computer vision based solutions could be a few millimeters when the target is within one meter [32], but the depth accuracy degrades to a few centimeters for an operational range of 5 meters [16]. The key limitation of computer vision based solutions is that the accuracy is highly dependent on the viewing angle and lighting conditions. Moreover, users may also have privacy concerns for video-camera-based solutions. Sound waves can be used to measure moving distances [23, 38, 39] or moving speeds [11]. When the user is holding the

device, sound wave based solutions can provide distance measurement accuracy of a few centimeters [23, 38]. Due to the weakness of sound energy reflected by hand, device-free gesture recognition solutions mostly use the Doppler effect, which only provides low-resolution speed measurements that cannot be used for fine-grained control over a long distance [11]. Recent fine-grained tracking solution only works for a short distance of 50 cm [21, 30]. QGesture uses the similar phase based distances measurement algorithm as LLAP [30]. However, our long-range WiFi gesture tracking system needs to handle the phase noises and interferences from nearby movements, which can be ignored in short-range sound-based systems.

3 SYSTEM MODEL

In this section, we first present the theoretical model that quantifies the gesture movement distance and direction. We then discuss the noise sources that make CSI measurements from COTS devices deviate from theoretical models. Finally, we present methods to remove the CFO and SFO in CSI measurements so that we can measure the movement distance and direction using theoretical models.

3.1 Modeling Phase-Distance Relationship

The magnitude and phase changes in CSI are closely related to the distance and direction of gesture movements. For simplicity, we first consider signals traveling through only two paths, *i.e.*, the Line-Of-Sight (LOS) path (path A) and the hand-reflected path (path B), between a pair of transmitter/receiver as shown in Figure 2. In theory, the resulting Channel Frequency Response (CFR) $H(f,t)$ in CSI measured at time t can be represented as [29, 34]:

$$H(f,t) = a_A(f,t)e^{-j2\pi f\tau_A(t)} + a_B(f,t)e^{-j2\pi f\tau_B(t)}, \tag{1}$$

where j is the imaginary unit with $j^2 = -1$, f is the frequency of the WiFi signal, $a_A(f,t)$ and $a_B(f,t)$ represent the magnitude attenuation and the initial phase in path A and B. As the path length of path A and B are different, their propagation delay $\tau_A(t)$ and $\tau_B(t)$ are also different as we have the relationship $\tau_A(t) = l_A(t)/c$, where $l_A(t)$ is the length of path A and c is the speed of light.

The CFR $H(f,t)$ contains two components: one static component for path A and one dynamic component for path B, as shown in Figure 3. Furthermore, the magnitude of the static component of different pairs of antenna of different subcarriers is different as a result of different propagation delay and different carrier frequencies as showed in our Section 5.5. Note that the CFR $H(f,t)$ is a complex value, where the real and imaginary part are called the In-phase (I) part and Quadrature (Q) part, respectively. Therefore, when we plot CFR in the complex plane, the CFR value at each time instance will be a vector and the end of the vector draws an I/Q trace as time evolves. In case that the hand pushes towards the transmitter/receiver, the I/Q trace for a single subcarrier is an arc as shown in Figure 3. This is because when the hand moves, the vector for path A, which is $a_A(f,t)e^{-j2\pi f\tau_A(t)}$, is not changed as both the transmitter and the receiver remain static. The vector for path A is a static component. However, the vector for path B, which is $a_B(f,t)e^{-j2\pi f\tau_B(t)}$, significantly changes when the path length of $l_B(t)$ changes. When $l_B(t)$ reduces, the attenuation $a_B(f,t)$ only changes slowly and the phase $\phi_B(t) = -2\pi f\tau_B(t) = -2\pi f l_B(t)/c$ increases significantly. For WiFi signals at 5 GHz, the radio wavelength $\lambda = c/f$ is equal to 6 cm. Therefore, the phase for the vector corresponding to path B, which is $\phi_B(t) = -2\pi l_B(t)/\lambda$, will increase by 2π when $l_B(t)$ is reduced by the radio wavelength of 6 cm. By measuring the phase change $\Delta\phi_B$ of the dynamic component, we can get the movement distance d as:

$$d = -\frac{\Delta\phi_B\lambda}{2a\pi}, \tag{2}$$

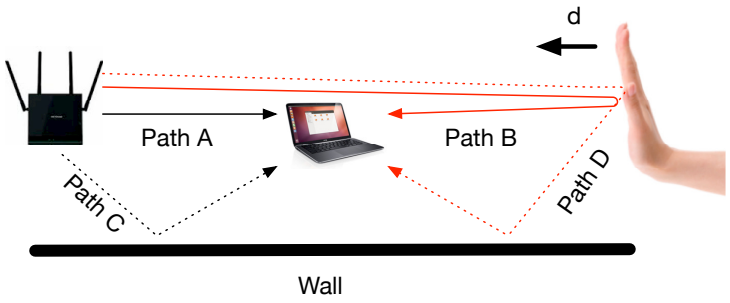


Fig. 2. Scenarios with multiple paths.

where a is the ratio of the path length change to the movement distance. For example, we have $a = 2$ for the scenario in Figure 2. Thus, we can measure the movement distance with an accuracy of a few centimeters. Furthermore, we can determine the movement direction by looking at the sign of $\Delta\phi$.

3.2 Practical Issues

We need to consider three practical issues before we can fit the above theoretical model into the real CSI measurements.

Static Multipath between Transmitter and Receiver: In indoor environments, the wireless signal will be reflected by surrounding objects such as walls, furniture, or doors. In case that the reflectors are static, *e.g.*, path A and C in Figure 2, we can treat the combination of these static paths as a single static component. Consider the case that there are n paths in total, where the n^{th} path is the dynamic path and other paths are static paths. Eq. (1) can be rewritten as:

$$H(f,t) = \underbrace{\sum_{i=1}^{n-1} a_i(f,t)e^{\frac{-j2\pi l_i(t)}{\lambda}}}_{\text{static component}} + \underbrace{a_n(f,t)e^{\frac{-j2\pi l_n(t)}{\lambda}}}_{\text{dynamic component}}. \tag{3}$$

Because the signal strength $a_i(f,t)$ and path length $l_i(t)$ of static component are constant in Eq. (3), all static multipaths can be treated as a single path with the corresponding I/Q vector as the sum of vectors for all static paths.

Dynamic Multipath through Moving Reflector: The signal reflected by the moving reflector could also travel through multiple paths, such as paths B and D in Figure 2. In this case, all these paths will have time-varying path lengths when the reflector moves. These time-varying paths will generate multiple time-varying components in the received signal strength. In practice, these components do not significantly affect the measurement accuracy, since the received reflecting signal strength is dominated by the direct reflection path, *e.g.*, path B in Figure 2. Human hands have small areas and the signals reflected by hands are weaker than the signal traveled through the static multipaths. Therefore, if the signal is reflected for more than once, *e.g.*, by both the hand and wall as in path D, the signal strength will be further attenuated. These multipath components will be small compared to the direct reflection path so that we can ignore them. As we will show in our experiments, the measurement error of QGesture only slightly increases when there are strong dynamic multipaths, *e.g.*, when pushing close to a wall.

Noises in CSI Measurements: Commercial CSI measurements contain various types of noises [29, 34]. The phase of CSI measurements is especially noisy due to the existence of Carrier Frequency Offset (CFO) and Sampling Frequency Offsets (SFO). The hand movement only changes the phase of the CSI by a small amount, *e.g.*, θ_1 in Figure 3. Therefore, without the accurate phase information, we can only gain a rough distance measurement

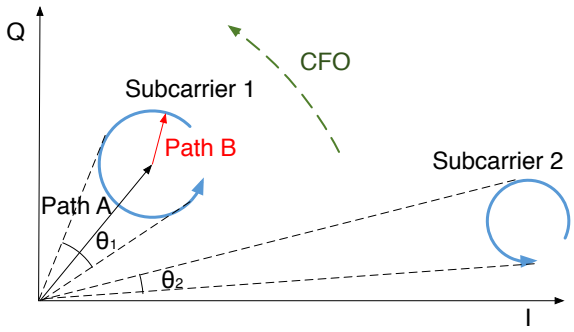


Fig. 3. I/Q Phasor Representation.

using the CSI magnitude as in [29], where the movement direction information has been lost. Furthermore, the magnitude of CSI measurements are also noisy due to the measurement errors. So, we need to first remove the phase and magnitude noises in CSI measurements before applying our theoretical model.

3.3 CSI Noise Sources

There are three types of noises in CSI measurements. The first two are phase noises and the last one is magnitude noise.

Carrier Frequency Offset (CFO):. Due to the small frequency differences in the carrier at the transmitter and receiver, the received CSI $H'(f,t)$ contains an extra phase shift as $H'(f,t) = e^{-j2\pi\delta f t}H(f,t)$, where δf is the CFO between the sender and the receiver [27, 29, 34]. As IEEE 802.11n standard allows the CFO between the transmitter and receiver to be as large as 100 kHz, the CFO introduces large phase uncertainties in CSI measurements. For example, when the CSI is measured at a rate of 4,000 frames per second, the phase change caused by CFO could be 50π between the two consecutive frames that are separated by a small time interval t of just 0.25 ms. Thus, the phase change caused by CFO is much larger than the phase changes caused by the movements, which changes by less than 1 radian during hand movements, as shown in Figure 3. Even a small measurement error in the time interval between consecutive frames will lead to a large phase offset, which makes the phase of CSI appear random.

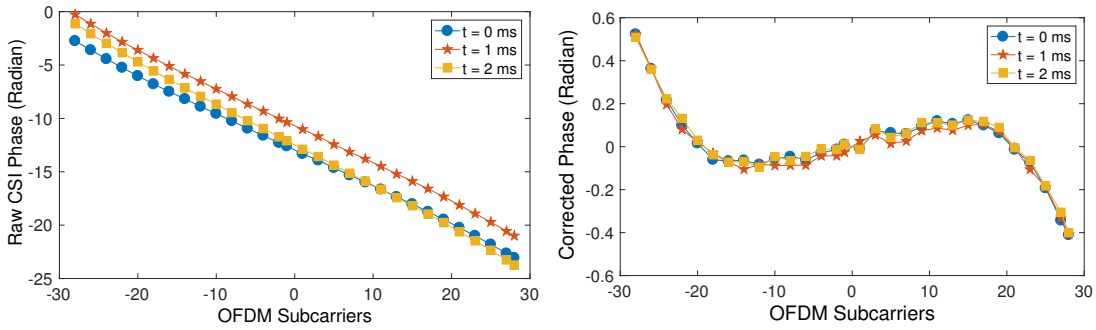
Sampling Frequency Offset (SFO) and Packet Boundary Detection (PBD) Error: These two error sources have similar effects which add another phase offset on the CSI as $H''(f,t) = e^{-j2\pi k\varphi}H'(f,t)$, where k is the index of OFDM subcarrier and φ is the phase offset [34]. Unlike the phase offset caused by CFO, which accumulates over time, the phase offsets of SFO and PBD have a linear relationship over different OFDM subcarriers. Moreover, the SFO/PBD offset has different slopes of φ on different frames.

Magnitude Variations: Due to the variations in transmission power and environmental noises, the magnitude of the CSI measurements also has large variations. These magnitude variations often have high energy impulses that could bury the small magnitude changes caused by hand movements [29].

3.4 Denoising CSI Measurements

We denoise CSI measurements using two steps:

1. Phase Correction: As discussed in Section 3.3, there are CFO, SFO and PBD phase offsets in CSI measurements. Fortunately, CSI values are simultaneously measured on 30 OFDM subcarriers for each antenna pair of the transmitter/receiver. For a transmitter with 2 antennas and a receiver with 3 antennas, we obtain $2\times3\times30 = 180$ CSI values for each WiFi frame. We can utilize the redundancy in CSI measurements to perform phase correction.



(a) Raw CSI phase. (b) Denoised CSI phase.

Fig. 4. The effect of phase correction on frames at different time instances.

We first remove the highly dynamical phase offset introduced by CFO. To estimate the value of CFO, we first observe that the phase of CSI measurements is mainly determined by three factors: CFO, SFO/PBD, and hand movements. As shown in [27], the phase offset of SFO and PBD is zero on the subcarrier with an index of $k = 0$. Therefore, the phase of subcarrier 0 for each antenna pair only contains the phase of CFO and the impact of hand movements. However, removing the phase of a randomly selected antenna pair could distort the small phase changes caused by hand movements. To preserve the impact of hand movements, we observe that the phase changes caused by the hand movement on different subcarriers are different as shown in Section 5.5. Consider the two subcarriers in Figure 3. Although the magnitudes of the dynamic components are similar for the two subcarriers, the phase change θ_2 in subcarrier 2 is much smaller because the magnitude of static component for subcarrier 2 is much larger than that for subcarrier 1. In real CSI measurements, we observe that there are subcarriers where the magnitude of static components is more than ten times higher than that of the dynamic component as shown in Section 5.5. In such cases, the phase change caused by hand movements (*e.g.*, θ_2 in subcarrier 2) in these subcarriers is smaller than 0.1 rad, which is an ignorable offset in other subcarriers (*e.g.*, θ_1 in subcarrier 1). Therefore, we can pick the CSI phase in subcarrier 0 of one antenna pair that has the largest magnitude of static components to serve as CFO reference. Based on these observations, we remove the CFO offset as follows. We first estimate the magnitude of the static component by taking long-term average on the CSI magnitude of each antenna pair. We then select the antenna pair with the largest CSI magnitude as the reference. Since the CSI is not measured on subcarrier 0, we interpolate the phase of subcarrier -1 and 1 to get CSI phase of subcarrier 0 of the selected antenna pair, which serves as CFO reference. We then subtract the calculated CFO in the phase of all subcarriers of other antenna pairs.

After removing the CFO, we correct the SFO/PBD offset, $-2\pi k\varphi$, for the remaining subcarriers based on a standard algorithm [17]. Note that the slope φ is different for every WiFi frame. Therefore, we perform a linear regression on the 30 subcarriers of each antenna pair to estimate the slope φ. We then remove the SFO/PBD for each subcarrier using the estimated φ. Figure 4 shows the CSI phase for one antenna pair before and after the phase correction. We observe the raw CSI phase changes significantly over a short duration of 1 ms as shown in Figure 4(a). The phase of the same subcarrier can differ by an amount of π and the slope of φ is also different at different time instances. The corrected phase in Figure 4(b) is more consistent over time. Furthermore, we can observe the small phase changes over different subcarriers after phase correction.

2. Magnitude Correction: After phase correction, we reduce the magnitude noise using oversampling. We measure the CSI value at a high sampling rate of 2,500 samples per second. Note that normal human movements introduce CSI magnitude variations with frequencies in the range of 1~100 Hz [29]. Therefore, we can use a low

pass moving average filter with window size of 80 samples to smooth the CSI magnitude, as well as the residual phase noises.

After the phase correction and magnitude correction, the CSI value of highly noisy subcarriers could still be corrupted. So we use subcarrier selection [20] and linear regression to reduce the error.

4 SYSTEM DESIGN

After removing the noises in CSI measurements, QGesture first uses the LEVD algorithm to remove the static component in the complex plane. We then measure the hand movement distance and direction based on the phase-distance relationship introduced in Section 3.1.

4.1 Real-world CSI Measurements

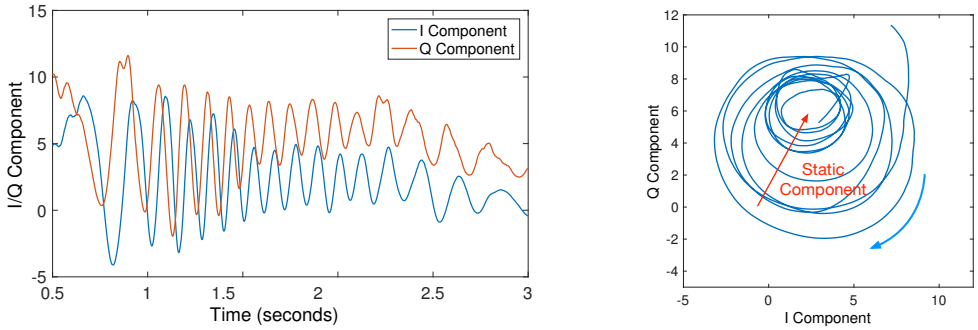
To better understand the CSI measurements provided by COTS devices, Figure 5 plots the CSI value captured by Intel 5300 network card. After denoising, the I/Q waveforms from COTS devices fit our theoretical model quite well. As the user pulls back his hand by 40 cm, we observe around 16 peaks in the waveform, which indicates that the phase is changed by 32π. Using the wavelength of 5.15 cm at 5.825 GHz, we get a path length change of 82.4 cm, which is very close to our model as we have $a = 2$ in this case. Furthermore, we observe that the phase is reducing (CSI values circling clockwise as in Figure 5(b)), which indicates that the user is pulling back.

We further observe that the real-world CSI values deviate from the theoretical model in two aspects. First, the static component is not a constant where the center of the circles in Figure 5(b) slowly changes. This is mainly due to the slow movements of the other body parts, such as the arms or the torso, of the user. Second, the magnitude of the dynamic component also changes. This is due to the reduction of strength of the reflected signal when the hand moves away from the transmitter/receiver. The slowly changing static component and reflected signal strength make it challenging to measure the phase of the dynamic component. For example, if we use a constant static component estimation, *e.g.*, with I component equal to 2, the last few small peaks in Figure 5(a) could be ignored and we will underestimate the movement distance.

4.2 Removing Static Components

QGesture uses a Local Extreme Value Detection algorithm (LEVD) to trace the slowly changing static component. The LEVD algorithm first initializes the static component estimation $S(t)$ as the long-term average, *e.g.*, average value over 2 seconds, of the CSI real part or imaginary part. As the channel coherent time for our WiFi scenario is about 10 *ms*, averaging over a time period of 2 seconds is enough to smooth out the channel variations. The algorithm uses an empirical threshold of T, which is determined by experiments, as shown in Section 5.5, to detect local maxima and minima. Once the CSI value deviates from its mean value by more than T, the algorithm starts detecting local maxima and minima. The local maxima and minima must satisfy the following two properties: 1) Local maxima must be at least larger than the current static component estimation $S(t)$ by the value of T. Similarly, local minima must be smaller than $S(t) - T$. 2) Local maxima and minima must appear alternately. If there are two consecutive local maxima/minima, we only retain the larger/smaller one.

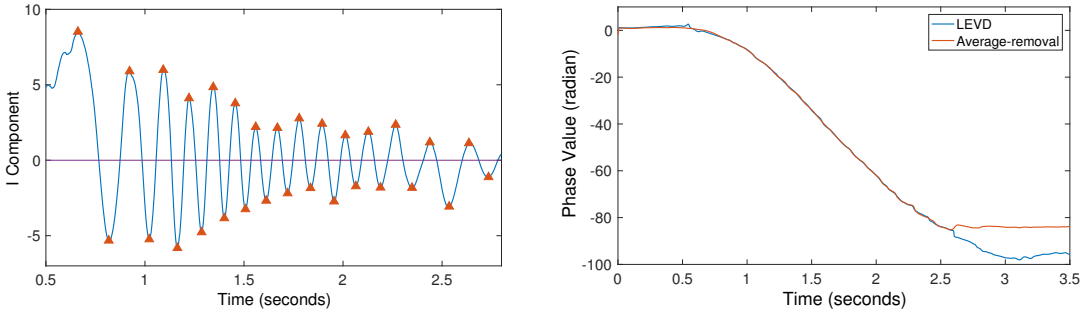
While detecting the local extrema, we update the static component estimation $S(t)$ dynamically by setting it to be the average of the last pair of local maximum and minimum values. In this way, LEVD is able to trace the slowly changing static component. Figure 6(a) shows the local extrema detected by LEVD, which precisely indicates the cycles of the waveform. After removing the estimated static component, LEVD gives a good estimation of dynamic component of the CSI waveform in Figure 6(a). The result of LEVD is better than simply removing the average of the CSI waveform. Figure 6(b) shows the distance estimation of LEVD and the simple average-removal algorithm for the changing path length of 80 cm. Using Eq. (2), we find that the pushing distance estimation of LEVD is 40.17 cm, while the distance estimation of simple average-removal is 33.6 cm. We observe that the



(a) I/Q waveforms. (b) I/Q trace in complex plane.

Fig. 5. Denoised CSI waveforms of pulling 40cm backward.

simple average-removal algorithm underestimates the distance as it ignores the last few small peaks in Figure 5(a).

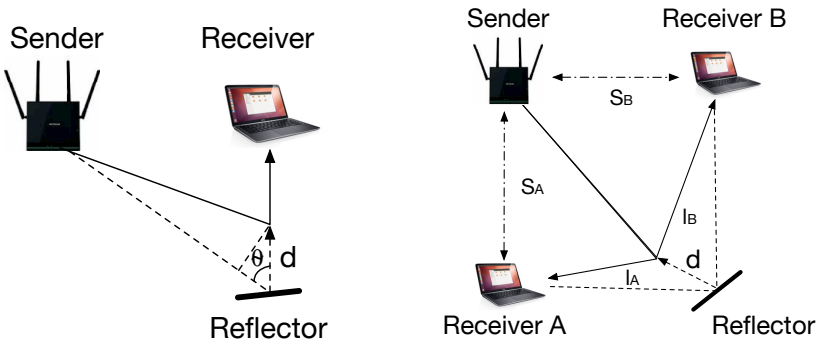


(a) Real part after LEVD. (b) Phase of two algorithms.

Fig. 6. Static component removal and phase measurements.

4.3 Measuring Movement Distance and Direction

After removing the static component, we use two different strategies to fuse the CSI values in different subcarriers and antenna pairs for movement distance/direction measurement. The first measurement method is called Principal Component Identification (PCI) [26]. In this method, we select the subcarrier that is most sensitive to the hand movement, *i.e.*, has the largest changes in phase during the movement, and use this subcarrier for measurements. The second measurement method uses Principal Component Analysis (PCA) on the *magnitude* of the CSI values over different subcarriers, in a similar way as in [29]. We select the second-largest PCA component and use Hilbert transform to recover the phase information from the magnitude and then use the recovered phase information to calculate the movement distance. These two algorithms have different advantages. The PCI algorithm retains the phase of CSI value so that we can determine the movement direction using this method. It works well when the hand is close to the sender/receiver. The PCA algorithm works better when the hand is more than 1.5 meters away from the sender/receiver. This is because the PCA algorithm "amplifies" the small changes caused by hand movement using the correlations in multiple subcarriers. However, as the PCA algorithm removes the phase information before processing, we cannot directly use it to detect the movement direction, which depends on whether the phase is increasing or decreasing.



(a) Different path length changes. (b) Multiple receivers.

Fig. 7. System setup for 2D hand tracking.

4.4 2D Hand Tracking

The path length change caused by a movement is determined by both the distance and the direction of the movement. Consider the case in Figure 7(a), where the reflector is not on the same line with the transmitter/receiver as shown in Figure 2. In this case, when the reflector moves by a distance of d towards the receiver, the path length of the reflected signal changes by a distance of $d + d\cos\theta$, where θ is the angle between the two segments of the reflection path. Therefore, the ratio of the path length change to the movement distance, defined as a in Eq. (2), changes with the position of the user. We use multiple receivers to allow arbitrary pushing directions. As shown in Figure 7(b), a movement with a distance of d will incur different path length changes at the two paths from the sender to two receivers A and B. When the distance between the sender and two receivers are known, we are able to localize the hand by intersecting two ellipses with foci at the corresponding sender and receivers [30]. Thus, we can determine both the movement angle and distance of the hand.

4.5 Preamble Detection

To separate gestures from background actions, QGesture defines "preamble" actions that mark the starting of the gesture. To activate QGesture, the user must perform a preamble action. Once QGesture detects the preamble action, it will start measuring the following pushing distance and direction and react to the movement information based on the measurements.

We use two gestures: *punch* as shown in Figure 8 and *double punch*, which is punching twice quickly with an interval shorter than 0.5 seconds, as the preambles. We choose these two gestures because they are easy to be performed in a short time, *i.e.*, less than 3 seconds. Moreover, the punch action has very distinctive characters as shown in Figure 8. They start with a quick movement of around 10 cm, which contains 3–6 cycles in CSI waveform. Then, there is a short pause of 0.2–0.4 seconds, followed by another movement of drawing the hand backward, which also contains 3–6 cycles in CSI waveform. Therefore, the punch gesture can be detected efficiently. Note that it is possible to use other activities, including voice or gaze, as preambles, which we leave as our future work.

QGesture uses the result of LEVD to detect the punch gesture. Each detected extreme point is classified into two classes, valid extrema and invalid extrema, using a logistic regression classifier. We use three features in the logistic regression, the absolute value of the current extremum, the absolute value of the previous extremum, and the time between the current and previous extremum. The regression model is trained using human labeled

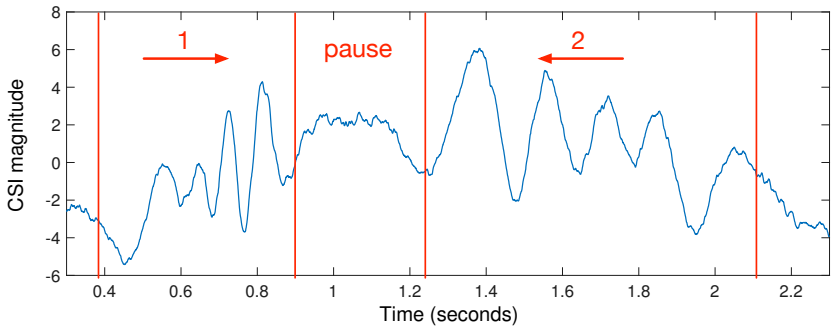


Fig. 8. CSI waveform for the punch gesture.

punch data, which contains 400 valid extremas for punch and 400 invalid extremas for extrema generated by other activities. The regression model calculates the probability that the detected extrema points belong to a punch action, based on its amplitude and intervals between neighboring extrema. In this way, we can detect punch gesture by searching for two segments with consecutive 5–12 valid extrema, separated by a short pause of 0.2–0.4 seconds. Using these simple criteria, QGesture can efficiently detect a punch or double punch with an accuracy higher than 92% and a false positive rate lower than 3%.

The preamble gestures serve for two purposes. First, they mark the starting point of the gesture so that other actions are not confused as gestures. Second, they indicate the user's intension for the following action. For example, we can use a single punch preamble to indicate that the following pushing action is to change the volume of the TV and use a double punch preamble to indicate an action of changing the brightness of the light. Note that we can introduce more gestures, such as snapping fingers, waving hands, as preambles to represent different meaning of the following gesture.

5 IMPLEMENTATION AND EVALUATION

5.1 Implementation

We implemented QGesture on COTS WiFi devices. QGesture has three components: A WiFi router (NetGear JR6100) which is configured to send 2,500 UDP packets every second, one or two ThinkPad X200 laptop with Intel 5300 wireless card, and a server for data processing. In the 1D scenario, QGesture use one laptop to collect per frame CSI at a rate of 2,500 samples per second using the Linux CSI tool [12]. In the 2D scenario, QGesture collect two separate CSI streams from two laptops placed in a setup as shown in Figure 9(b). The clocks of these two laptops are synchronized by NTP so that we have synchronized timestamp for each CSI measurement. The collected CSI data are forwarded to a server which performs signal processing using MATLAB.

5.2 Experimental Environments

Figure 9 shows the experimental environments for the performance evaluation in the 1D scenario and the 2D scenario. We performed the experimental evaluation in three different environments. The first one was a small conference of size 7.5 × 6.5m with a table of 2 × 2m, with the sender/receiver placed on the table. The second one was in the same room as the first one, but the transmitter and the receiver were placed near the wall which introduced rich multipath interferences. The third one was a large lobby area with a size of 20 × 20m. If not specified, most of our experiment was performed in the first environment. If not specified, the sender and the receiver were placed on a table with a height of about 1 meter. In the 1D scenario, the default distance between the sender and the receiver was 1 meter. For 2D hand tracking, the distance between the sender and the two

(a) 1D Scenario. (b) 2D Scenario.

Fig. 9. Evaluation scenario in small conference room.

receivers were 1 meter and 1.2 meters, respectively. We use an asymmetry setup in 2D tracking for asymmetry settings usually provide better performances.

During the experiments, we used the similar WiFi channel settings as our campus network. If not specified, WiFi channel 165 (f=5.825 GHz) with 40MHZ bandwidth was used. We chose this channel for the shorter wavelength in 5 GHz bands which leads to better sensitivity to hand movements. On average, there were 5 other devices using the same channel in the campus network when we performed the experiment. Hence, QGesture is robust to interference from other WiFi devices using the same channel.

5.3 Experimental Methodology

Our experimental data was collected under IRB approval (IRB#15-1042). Five volunteers were involved in the data collection process during a period of three weeks. The volunteers were 3 males and 2 females with ages between 22 – 25. All volunteers were graduate students with engineering backgrounds. The volunteers were given short 10-minutes introductions of the systems and instructions of the experiments before the experiment. For each experiment, the volunteers were allowed to practice a few times to ensure that they correctly followed the instructions and the gesture recognition system was operating correctly.

During the experiments, volunteers were instructed to push their hand normally and naturally either standing close to the receiver or sitting on a chair close to the receiver. If not specified, the distance between the volunteer to the receiver is 1 meter. In the 1D scenario, the volunteers were instructed to push their hand along the same line of the transmitter and receiver. In the 2D scenario, the volunteers were instructed to push along different trajectories, such as a straight line with an angle of 30 degrees, 45 degrees, and 60 degrees towards the transmitter. If not specified, each volunteer was instructed to push his/her hand for 50 times for each experiment. The ground truth of the movement distance was measured with a ruler placed along the arm of the volunteer. When the subject was conducting the experiment, other subjects and the experiment coordinator could be in the same room observing the movement and reading the ground truth of the movements.

5.4 Evaluation Metrics

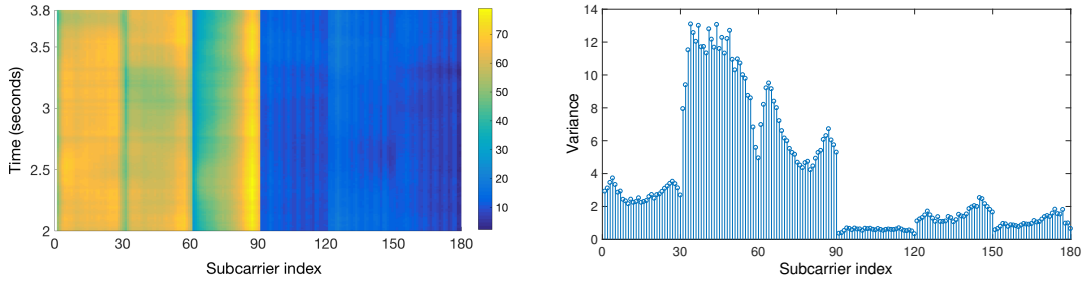
We evaluated QGesture from the following three perspectives: effectiveness, robustness, and efficiency. For effectiveness in the 1D scenario, we evaluated the movement distance measurement accuracy at different operational distances. For effectiveness in the 2D scenario, we evaluated the movement direction and distance measurement when the arm is moving along a line with a predefined angle with respect to the sender and receiver. We also evaluated the performance of preamble detection in terms of recognition accuracy and False Positive Rate (FPR). The recognition accuracy for preamble detection is defined as the number of detected preambles divided by the total number of preambles performed. The FPR for preamble detection is defined as the number of falsely detected preambles divided by the number of non-preamble activities performed. For robustness, we

evaluated the performance of QGesture on different persons, different body parts, and different environments. For efficiency, we evaluated the processing time of the system and the impact on co-existing WiFi users.

5.5 Effectiveness

We first verified our assumptions that different pairs of antennas receive different magnitudes of static components as described in Section 3.4. Figure 10(a) shows the raw CSI data pattern of pushing hand in our 180 subcarriers of which the first 60 subcarriers correspond to the first transmitting antenna. We observe that different antenna pairs have different powers. The power is mainly from the magnitudes of static components because of the small reflected area of our small hand. Therefore, we can use the pair of antenna that has the biggest magnitudes of static components to remove the impact of CFO while keeping the phase information of hand movement. We also observe that the magnitude of the static component is more than ten times higher than that of the dynamic component combining Figure 10(a) and 10(b) which is mainly the magnitude of dynamic component.

The effectiveness of QGesture is evaluated in both LOS scenario and NLOS scenario in a small conference room as shown in Figure 11. In both scenarios, the user performs gesture at the position marked with a star. We change the distance D_u between the user and receiver and the distance D_t between the transmitter and receiver, to evaluate the performance of QGesture in different cases.

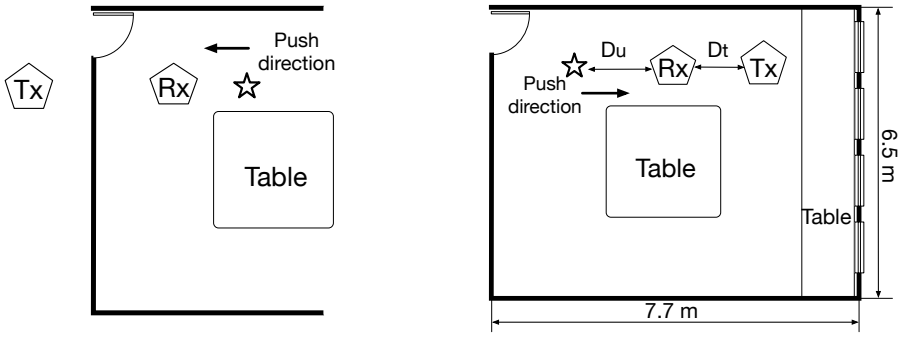


(a) CSI time series pattern of all pairs of antennas.

(b) Variance of different subcarriers.

Fig. 10. Example of different magnitudes of static component on different subcarriers.

1D Pushing Distance Measurement Accuracy: *QGesture can measure the distance of hand movement with a 90th percentile measurement error of less than 4 cm at a distance of 1 meter.* To verify the performance of our phase correction algorithm, we use different methods to remove the impact of CFO. One is that we chose the subcarrier that has the largest amplitude as reference subcarrier (Largest-PCI), and the other is that we randomly selected the reference subcarrier (Random-PCI). Moreover, we use Hilbert transformation on the CSI stream after PCA processing to get the phase information to measure the moving distance (Hilbert-PCA). Figure 12(a) shows the distance measurement error CDF for a volunteer pushing 80 times for a distance of 40 cm, which is around the longest pushing distance of an adult, at a distance of 1 meter to the receiver. We observe that the average distance measurement error is 2.21 cm, while for 90% cases the error is smaller than 4 cm using Largest-PCI. However, the average distance measurement error is 4.16 cm using Hilbert-PCA, because the phase information given by Hilbert transformation is not as accurate as that given by our phase correction algorithm. In addition, the average distance measurement error is 7.42 cm using Random-PCI, this is because that randomly selecting subcarrier as reference subcarrier may also remove the phase information caused by hand movement. For shorter pushing distances as shown in Figure 12(b), such as 10 cm, the average measurement error reduces to less than 1



(a) NLOS scenario. (b) LOS scenario.

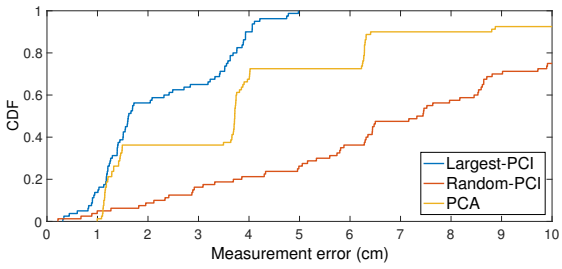
Fig. 11. Experiment setups.

cm for Largest-PCI. This is because that the shorter the pushing distance, the smaller accumulated measurement distance error. In the following experiment, we call Largest-PCI as PCI and Hilbert-PCA as PCA for short.

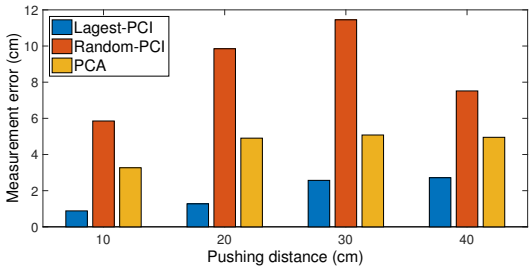
1D Pushing Direction Accuracy: *QGesture can reliably detect the pushing direction with accuracy more than 95% with the latency of 0.2 seconds.* Figure 14 shows the movement direction accuracy of 5 different people pushing at a distance of 1 meter to the receiver. The detection accuracy increases when we use longer pushing samples for detection. For all users, the direction can be determined with more than 95% accuracy when using a pushing sample of 0.2 seconds.

2D Pushing Direction and Distance Accuracy: *At a distance of 1 meter, the overall mean absolute direction error of tracking the straight line in different directions in a normal conference is 15.19 degrees and the overall mean measurement error in distance is 3.75 cm for pushing 40 cm.* Figure 9(b) shows our two dimensional scene setting depicted as Section 4.4. Figure 13 shows the tracking result of a volunteer pushing 57 cm with an angle of 45 degrees. The initial location is measured beforehand. We track the hand location with a time interval of 19.2 ms. In each time interval, we use the linear regression of time and phase on one channel to remove small phase error after our correction algorithm. The tracking direction is determined by the sign of the slope of the regression line of both receivers. The tracking distance is determined by the phase change of the regression line of both receivers. Figure 15(a) shows the error of movement direction of four different scenarios. The four scenarios contain 8 different samples where the volunteer pushes his/her hand along these paths forth and back. Our absolute direction error is calculated in each time interval. Our pushing direction accuracy is similar to those in WiDir [33], which tracks the walking direction. However, the walking human tracked in WiDir has bigger reflection area than the hand tracked in QGesture. This indicates that utilizing our phase information after removing phase offset is more accurate than utilizing amplitude information. Figure 15(b) shows the distance error of pushing 40 cm of the corresponding direction. The distance of these different directions is calculated by the distance between the ending point and the starting point.

Operational Distance: *QGesture achieves distance measurement accuracy of smaller than 5.5 cm within distance of 2 meters.* Figure 16(a) shows the average amplitude of CSI variations caused by hand movement with different distances between the hand and receiver (D_u in Figure 11(b)). We observe that the amplitude of CSI variations reduces when the distance between the user to the receiver increases. Our experiments show that the average amplitude of CSI variation during the silent period is 0.71 with a standard deviation of 0.46. Based on these measurements, we set the empirical threshold T used in LEVD algorithm to 2, which is three times of the standard deviation away from the mean, so that it can reliably detect gestures performed at 2 meters away.



(a) CDF of pushing 40 cm.



(b) Average measurement error.

Fig. 12. Distance measurement accuracy.

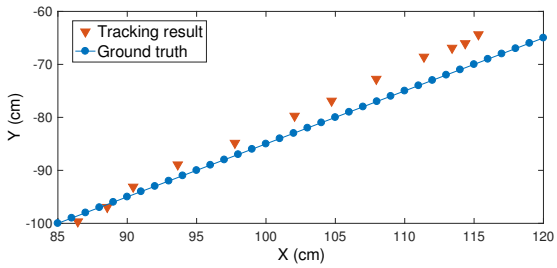


Fig. 13. Example of tracking result.

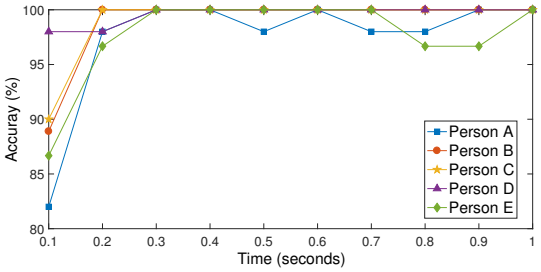
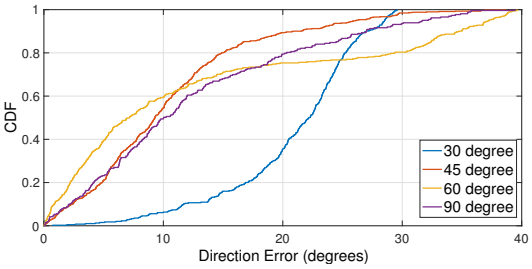
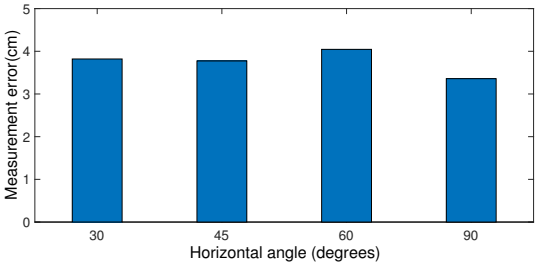


Fig. 14. Movement direction accuracy.



(a) CDF of absolute direction error.

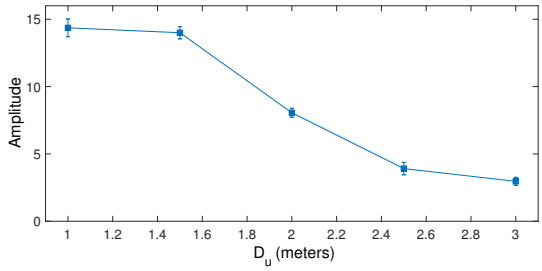


(b) Average measurement error for different directions.

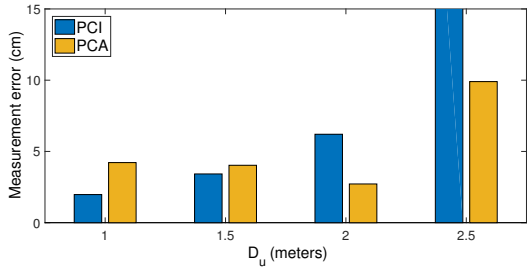
Fig. 15. 2D tracking results.

Figure 16(b) shows the distance measurement accuracy for pushing distance of 40 cm with different D_u, the distance between the user and the receiver, while D_t is fixed at 1 meter. We observe that the measurement accuracy decreases when D_u increases and at a distance of 2.5 meters the error of both PCI and PCA is larger than 10 cm. As D_u increases, the performance of PCI decreases quickly, while PCA can achieve less than 4 cm error when the hand is at 2 meters. This shows PCA works better than PCI on weak signals.

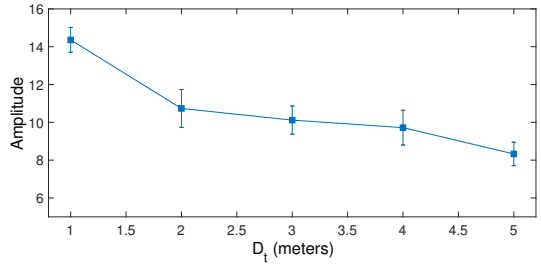
QGesture achieves measurement error smaller than 2.7 cm where there is no LOS between the transmitter and receiver. Figure 16(c) shows how the CSI variations changes with increasing distance between transmitter and receiver (D_t in Figure 11(b)). We observe that the amplitude only slightly decreases when the distance between transmitter and receiver increases. This indicates that the amplitude of the reflection is mainly determined by the distance between the reflector to the receiver. Figure 16(d) shows the distance measurement error with different



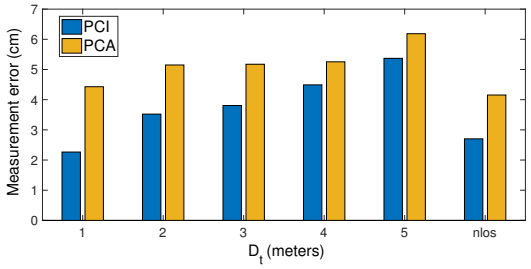
(a) CSI amplitude with different D_u.



(b) Measurement accuracy with different D_u.

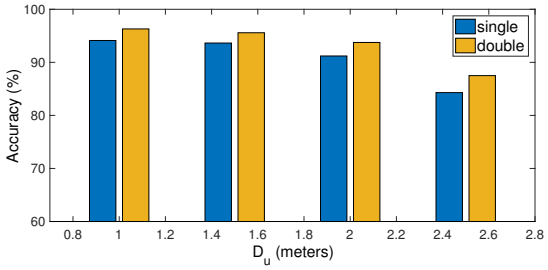


(c) CSI amplitude with different D_t.

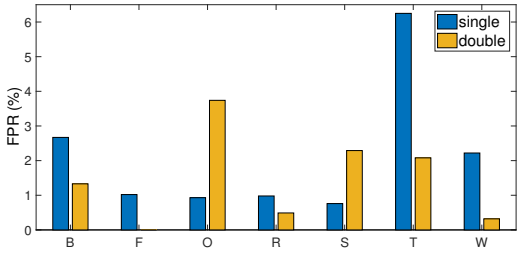


(d) Measurement accuracy with different D_t.

Fig. 16. Measurements of distance with different distance between user, transmitter, and receiver.



(a) Detection accuracy for single and double punch.

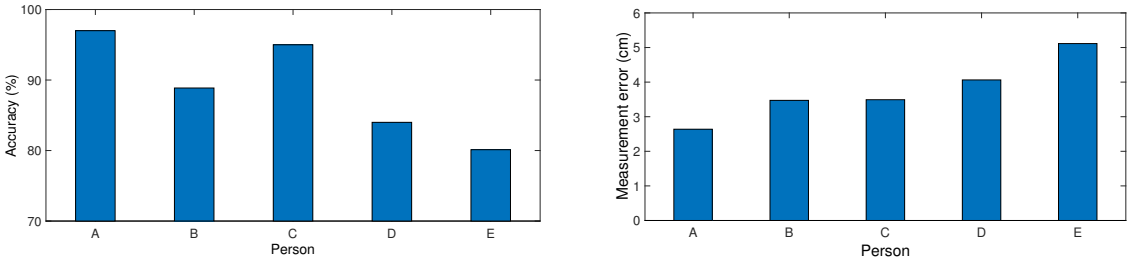


(b) FPR of confusing the preamble with seven activities.

Fig. 17. Preamble detection performance.

D_t. For the NLOS scenario in Figure 11(a), the distance measurement error is still less than 2.7 cm. PCI has better performance than PCA for different D_t values.

Preamble Detection: *At a distance of 2 meters, QGesture detects preamble gesture of single punch and double punch with accuracy of 91.2% and 93.5%, respectively.* Figure 17(a) shows the detection accuracy for single and double punch at different distances. We observe that the accuracy for detection is higher than 90% within 2 meters, while remains 84.3% and 87.5% at a distance of 2.5 meters. *QGesture has a low False Positive Rate (FPR) of 3.2% for preamble detection on daily activities.* To evaluate the FPR for daily activities, we collected an activity database containing 1,000 samples of seven activities, including Boxing (B), Falling (F), Opening Refrigerator (O), Running (R), Sitting down (S), Brushing Teeth (T), and Walking (W). Figure 17(b) shows the FPR of confusing the preamble with other activities. Most of these activities have FPR lower than 3%, except for brushing teeth, which is actually a regular forth-back movement similar to the punch gesture.



(a) Preamble detection accuracy.

(b) Average measurement error.

Fig. 18. Robustness over different persons.

5.6 Robustness

To evaluate the robustness of QGesture, we evaluated QGesture on five persons, including three males and two females with age in the range of 22 $\sim$ 25. Each person performs 20 gestures at different distances to the receiver and the measurement error is averaged over the 20 samples.

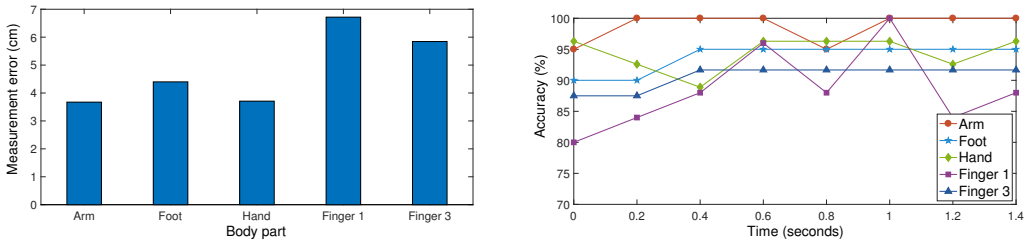
Robustness for Different Persons: *QGesture achieves average preamble detection accuracy of 89% and distance measurement error of 4 cm for different persons.* Figure 18(a) shows the preamble detection accuracy of the five subjects that perform the preamble at a distance of 1 meter to the receiver. The average preamble detection accuracy is 89%, while lowest accuracy over the five subjects is 80% (subject E). Note that the linear regression for preamble detection is only trained on data collected from subject A. Therefore, our preamble detection algorithm can be applied to persons that are not in the training set. The average distance measurement for a pushing distance of 30 cm with D_u equal to 1 meter is 4 cm, while the largest error is 5 cm (also subject E).

Robustness for Different Environments: *QGesture achieves distance measurement error of less than 5 cm in different environments.* Table 1 shows the distance measurement error for pushing distance of 30 cm in different environments. We observe that the measurement error is small in the last two cases when there are no large reflectors around. The small changes in the measurement error of small room and lobby are mostly caused by variations of the experimental results. The largest average measurement error is 4.44 cm when the transmitter/receiver is near the wall. This is due to the dynamical multipath caused by the wall as discussed in Section 3.2. *QGesture achieves average preamble detection accuracy of 89% in different environments.* Table 1 shows the preamble detection accuracy of single punch and double punch in different environments. We observe that the multipath effect of the wall slightly decreases the preamble detection accuracy from more than 92% in other environments to 89% in the "Near wall" scenario.

Robustness for Different Body Movements: To evaluate the granularity of QGesture, we ask one volunteer to move his different body part, including arm, foot, hand, one finger, and three fingers, to control QGesture. The deployment of the devices is the same as above experiments. Note that QGesture measures the specific gesture movement pattern rather than recognizing which body part is moving so that users may control the system using different body parts by performing similar movements as hand gestures. Figure 19 shows the performance for QGesture when measuring different body movements. We observe that larger areas of reflection for the moving body part leads to smaller measurement errors. For example, the average measurement error for single finger movement is about 7 cm while the average measurement error for arm movement is less than 4 cm. Furthermore, we can determine the arm movement direction with an accuracy of 100% with a latency of 0.2 seconds, while the accuracy of finger movement is as low as 80% as shown in Figure 19(b). This is because the reflection area of one finger is so small that the corresponding CSI fluctuation is too weak to be reliably measured by QGesture.

Table 1. Measurement error in different environments.

	Near wall	Small room	Lobby
Error (cm)	4.44	3.10	2.89
Single accuracy	89.1%	94.1%	93.5%
Double accuracy	90.3%	96.3%	93.8%



(a) Average measurement error

(b) Movement direction accuracy

Fig. 19. The performance of pushing different body parts.

5.7 Efficiency

We use MATLAB to process the CSI measurements, including phase correction, LEVD-based peak detection, and tracking. For offline evaluation, the overall processing time for 1 second of data only takes 0.7 seconds on a desktop with an Intel Core i7 CPU at 3.5 GHz. Therefore, QGesture can run in real-time on commercial desktop computers. For real-time processing, our algorithm operates on samples with a segment size of 750 samples, which incurs about 0.2 seconds delay in movement distance measurement and direction detection. Such latency is perceivable by human users, we plan to further reduce the latency our algorithm to less than 50 ms in the future.

To evaluate the impact on co-existing WiFi users' throughput, we measure the uplink and downlink TCP throughput of the one co-existing user as shown in Table 2. QGesture reduces WiFi throughput co-existing user by a moderate amount of 9.11% and 19.03% in term of uplink and downlink TCP throughput when the sampling rate of QGesture is 2500 samples/second.

5.8 Case Study

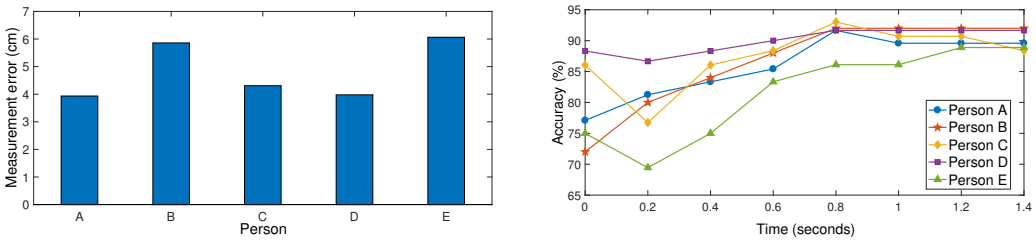
One of the most popular applications of our system is the smart home scenario as shown in Figure 20. Considers the case that Alice increased the volume of the TV when watching her favorite TV show. While her roommate, Jessica, is messaging on her mobile phone, the raised volume is too noisy for Jessica. Hence, Jessica wishes to decrease the volume of TV after finding Alice increased the volume of TV. With the in-air gesture, they can adjust the same device in turn in this scenario, without fighting for the remote controller. To evaluate the performance of our system in this scenario, we asked five persons to control the device in-turn in a small room that is similar to the LOS scenario described in Figure11(b).

Participants and Apparatus: In this experiment, the five volunteers are 3 females and 2 males with an average age of 24. All are first-time users to our system. We ask one of the five volunteers to sit down at 1 meter away from the receiver and push his/her hand forward and backward with a total number of 25 circles. Moreover, we require the other volunteers to sit down at any positions in the small room and not to perform large movements such as walking, running and pushing, they can talk to each other or read books while one volunteer is pushing his hand. These small movements normally do not interfere with QGesture. The five volunteers take turns to perform gestures. The experiment coordinator measures the length of the arm of each volunteer to

Table 2. TCP throughput of one co-existing WiFi user.

	With QGesture	Without QGesture
Uplink throughput (Mbps)	34.14	37.56
Downlink throughput (Mbps)	33.66	41.57

Fig. 20. The application scenario in smart home.



(a) Average measurement error.

(b) Movement direction accuracy.

Fig. 21. The performance of five persons perform gestures in-turn.

get the ground truth of the distance and record the ground truth of the pushing direction. The CSI streams are processed offline.

Results: *QGesture achieves distance measurement accuracy of smaller than 5 cm and can reliably detect the pushing direction with accuracy more than 90% with a latency of 1 second in the scenario that five persons push hand in-turn as shown in Figure 21.* Figure 21(a) shows the average measurement error of different persons pushing hand, where person B and person E have higher measurement error than others because the hand areas of the two persons are smaller. The latency for detecting the pushing direction is around 1 second for all users as shown in Figure 21(b). Moreover, because there are five persons in the small room, their movements may introduce errors in the phase measurements, which increases the time latency of detecting pushing direction.

6 LIMITATIONS

QGesture establishes the feasibility of using COTS WiFi devices for measuring pushing distance and detecting pushing direction. However, the current implementation of QGesture has two major limitations. First, the user must perform the two predefined actions, punch and double punch, as preambles so that a gesture instruction can be separated from daily activities. A more natural interaction would be allowed users to push towards the home appliance that he/she wishes to control, without an extra preamble. Furthermore, voice or gaze can also be used as preambles to improve the user experience. There are interesting directions for our future study. Second, when there are multiple users pushing at the same time, the received signal will be a complicated combination

of multiple dynamic multipaths. In this case, QGesture cannot reliably measure the movement distance and direction. Furthermore, QGesture cannot determine the gesture performer when there are multiple persons in the same room. In addition, if there are strong moving reflectors in the environment, QGesture also cannot measure the gesture movement distance with a high accuracy.

7 CONCLUSIONS

We make three key contributions in this paper: First, we explore the possibility to measure gesture movement distance and direction using COTS WiFi devices. Such fine-grained gesture measurements can lead to natural device-free interactions that emulate physical input devices such as knobs and levers. Second, we propose a suite of signal processing techniques and algorithms, such as the phase correction algorithm and the LEVD algorithm, to implement our quantifiable WiFi signal based gesture recognition system. Third, we implemented QGesture on COTS WiFi devices and conducted comprehensive evaluations. Our experimental results show that QGesture achieves an average accuracy of 3 cm in the measurement of movement distance.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China under Grant Numbers 61472185, 61472184, 61321491, 61502229, 61373130, 61672276 and 61373129, the National Science Foundation under Grant Numbers CNS-1318563, CNS-1524698, CNS-1421407, CNS-1565609, and IIP-1632051, Collaborative Innovation Center of Novel Software Technology and Industrialization, the Jiangsu High-level Innovation and Entrepreneurship (Shuangchuang) Program, China 973 projects (2014CB340303), and China NSF Grants 61672353,61472252,61321491.

REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. WiGest: A Ubiquitous WiFi-based Gesture Recognition System. In *Proceedings of IEEE INFOCOM*.

[2] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-Person Motion Tracking via RF Body Reflections. In *Proceedings of Usenix NSDI*.

[3] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 2013. 3D Tracking via Body Radio Reflections. In *Proceedings of Usenix NSDI*.

[4] Jake K. Aggarwal and S. Ryoo Michael. 2011. Human activity analysis: A review. *Comput. Surveys* 43, 3 (2011).

[5] Mohammed Abdulaziz Aide Al-qaness and Fangmin Li. 2016. WiGeR: WiFi-based gesture recognition system. *ISPRS International Journal of Geo-Information* 5, 6 (2016), 92.

[6] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *ACM MobiCom*.

[7] Xiaoxiao Cao, Bing Chen, and Yanchao Zhao. 2016. Wi-Wri: Fine-Grained Writing Recognition Using Wi-Fi Signals. In *Proceedings of IEEE Trustcom/BigDataSE/ISPA*.

[8] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Tracking Keystrokes Using Wireless Signals. In *Proceedings of ACM MobiSys*.

[9] Han Ding, Chen Qian, Jinsong Han, Ge Wang, Wei Xi, Kun Zhao, and Jizhong Zhao. 2017. RFIPad: Enabling Cost-Efficient and Device-Free In-air Handwriting Using Passive Tags. In *Proceedings of IEEE ICDCS*.

[10] Emre Ertin, Nathan Stohs, Santosh Kumar, Andrew Raij, Mustafa al'Absi, and Siddharth Shah. 2011. AutoSense: Unobtrusively Wearable Sensor Suite for Inferring the Onset, Causality, and Consequences of Stress in the Field. In *Proceedings of ACM Sensys*.

[11] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: Using the Doppler Effect to Sense Gestures. In *Proceedings of ACM CHI*.

[12] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM CCR* 41, 1 (2011), 53.

[13] Wenfeng He, Kaishun Wu, Yongpan Zou, and Zhong Ming. 2015. Wig: WiFi-based Gesture Recognition System. In *Proceedings of IEEE ICCCN*.

[14] Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. 2015. WiDeo: Fine-grained Device-free Motion Tracing using RF Backscatter. In *Proceedings of Usenix NSDI*.

[15] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing Gesture Recognition to All Devices. In *Proceedings of Usenix NSDI*.
[16] Kourosh Khoshelham. 2011. Accuracy analysis of kinect depth data. In *Proceedings of ISPRS workshop Laser Scanning*.
[17] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization using WiFi. In *Proceedings of ACM SIGCOMM*.
[18] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: Talk to Your Smart Devices with Finger-grained Gesture. In *Proceedings of ACM UbiComp*.
[19] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics* 35, 4 (2016), 142.
[20] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. 2015. Tracking Vital Signs During Sleep Leveraging Off-the-shelf WiFi. In *Proceedings of ACM Mobihoc*.
[21] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of ACM CHI*.
[22] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. 2014. Wi-Fi Gesture Recognition on Existing Devices. arXiv preprint arXiv:1411.5394. (2014).
[23] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. Beepbeep: a High Accuracy Acoustic Ranging System using COTS Mobile Devices. In *Proceedings of ACM SenSys*.
[24] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home Gesture Recognition Using Wireless Signals. In *Proceedings of ACM MobiCom*.
[25] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *Proceedings of ACM MobiCom*.
[26] Sheng Tan and Jie Yang. 2016. WiFinger: Leveraging Commodity WiFi for Fine-grained Finger Gesture Recognition. In *Proceedings of ACM Mobihoc*.
[27] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *Proceedings of USENIX NSDI*.
[28] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proceedings of ACM SIGCOMM*.
[29] 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 ACM MobiCom*.
[30] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of ACM MobiCom*.
[31] Teng Wei and Xinyu Zhang. 2015. mTrack: High-Precision Passive Tracking Using Millimeter Wave Radios. In *Proceedings of ACM MobiCom*.
[32] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors* 13, 5 (2013), 6380–6393.
[33] Dan Wu, Daqing Zhang, Chenren Xu, Yasha Wang, and Hao Wang. 2016. WiDir: Walking Direction Estimation Using Wireless Signals. In *Proceedings of ACM UbiComp*.
[34] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise Power Delay Profiling with Commodity WiFi. In *Proceedings of ACM MobiCom*.
[35] Lei Yang, Qiongzheng Lin, Xiangyang Li, Tianci Liu, and Yunhao Liu. 2015. See Through Walls with COTS RFID System!. In *Proceedings of ACM MobiCom*.
[36] Koji Yatani and Khai N Truong. 2012. Bodyscope: a Wearable Acoustic Sensor for Activity Recognition. In *Proceedings of ACM UbiComp*. 341–350.
[37] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. 2007. Challenges: device-free passive localization for wireless environments. In *ACM MobiCom*.
[38] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a Mobile Device into a Mouse in the Air. In *Proceedings of ACM MobiSys*.
[39] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. 2012. Swordfight: Enabling a New Class of Phone-to-phone Action Games on Commodity Phones. In *Proceedings of ACM MobiSys*.

Received November 2017; revised January 2018; accepted March 2018