WiTrace: Centimeter-Level Passive Gesture Tracking Using WiFi Signals

Lei Wang[1] Ke Sun[1] Haipeng Dai[1] Alex X. Liu[1,2] Xiaoyu Wang[1]
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China[1]
Dept. of Computer Science & Engineering, Michigan State University, East Lansing, MI, 48824, USA[2]
wangl@smail.nju.edu.cn, {kesun, mg1633074}@smail.nju.edu.cn, haipengdai@nju.edu.cn

***Abstract*—Gesture tracking is a basic Human-Computer Interaction mechanism to control devices such as electronic Internet of Things and VR/AR devices. However, prior WiFi signal based systems focus on gesture recognition and provide results with insufficient accuracy, and thus cannot be applied for high-precision gesture tracking. In this paper, we propose a CSI based device-free gesture tracking system, called WiTrace, which leverages the CSI values extracted from WiFi signals to enable accurate gesture tracking. For 1D tracking, WiTrace derives the phase of the signals reflected by the hand from the composite signals, and measures the phase changes to obtain the movement distance. For 2D tracking, WiTrace proposes the first CSI based scheme to accurately estimate the initial position, and adopts the Kalman filter based on Continuous Wiener Process Acceleration model to further filter out tracking noise. Our results show that WiTrace achieves the estimated accuracy of 3.91 cm for initial position on average, and achieves cm-level accuracy, with mean tracking errors of 1.46 cm and 2.09 cm for 1D tracking and 2D tracking, respectively.**

I. Introduction

A. Motivation

Gesture tracking is a basic Human-Computer Interaction mechanism to control not only electronic Internet of Things devices but also VR/AR devices. In smart home, gesture is recognized to change the channel of TVs or increase the temperature of air conditioners. For VR/AR, users use gesture tracking to interact with devices, such as writing words and typewriting in the air. Recently, WiFi signals are widely used for passive sensing for gesture movement as [1]–[3] due to its particular advantages. In comparison with vision based methods [4], [5], WiFi based approaches are not limited by lighting condition and room layout as WiFi signals are able to penetrate through walls. Meanwhile, users don't bother to wear devices [6], which is convenient and saves extra cost.

B. Limitations of Prior Art

Prior WiFi based gesture recognition systems extract features from reflected signals for different gestures [2], [7] and use machine learning methods to recognize gestures. Nevertheless, these methods provide results with insufficient accuracy and cannot be applied to high-precision gesture tracking. To the best of our knowledge, there are only two WiFi based tracking schemes, *i.e.*, WiDraw [8] and Widar [9]. In particular, WiDraw applies Angle-Of-Arrival (AOA) measurements to reach the accuracy of 5 cm of tracking. It has a limited working range of 2 feet and requires 25 WiFi transmitters, which inhibits usage in applications such as IoT ones. Moreover, Widar is a human tracking scheme which treats the human body as one single object with decimeter-level resolution. The dominant technologies above are as shown in Table I. Other Radio Frequency (RF) and acoustic signal based tracking schemes use localization technologies to track gestures. Based on 60 GHz wireless technology, Google uses Soli system [10] to track small gesture and mTrack [11] uses 60 GHz steerable and highly-directional beams to track the movement of a finger or a pen. However, the power of 60 GHz signal has a fast attenuation in the air. Thus, both of them are not suitable to serve as a remote control for smart home. WiTrack [12], [13] proposes to use specially designed Frequency-Modulated Continuous-Wave radar with a high bandwidth of 1.79 GHz to track human movement behind the wall with a resolution of about 11 cm to 20 cm, which needs special hardware. Similar to RF signal, although acoustic tracking schemes [14]–[17] have high accuracy, these systems cannot serve as remote control interface for home applications due to limited working range.

C. Proposed Approach

In this paper, we propose WiTrace, a WiFi based device-free cm-level gesture tracking system. Our key idea is to use the Channel State Information (CSI) values of WiFi to track the hand with centimeter-level accuracy in 2D space. We utilize the fact that the phase changes of CSI values reflected by the hand are proportional to the propagation path length changes of the hand. Since the wavelength of 2.4 GHz WiFi signals is around 12.5 cm, hand movement with a few centimeters will significantly affect the CSI values. WiTrace uses Universal Software Radio Peripheral (USRP) to transmit and receive the Commercial-Off-the-Shelf (COTS) 802.11g signals with frequency 2.4 GHz and bandwidth 20 MHz. For 1D tracking, WiTrace extracts the phase of the signals reflected by the hand from the composite signals, and measures the phase changes to obtain the movement distance. Furthermore, WiTrace uses one transmitter and two receivers to enable the 2D tracking of hand. We propose the first CSI based scheme to accurately estimate the initial position, which has huge impact on the overall system performance. Further, we adopt the Kalman filter based method to filter out noise of tracking.

D. Technical Challenges and Solutions

WiTrace addresses three critical challenges. The first challenge is to achieve cm-level hand tracking accuracy for large range based on WiFi signals. Prior WiFi based tracking scheme uses AOA to track hand with large number of transmitters in the range of 2 feet [8]. In contrast, we leverage the fact that the phase changes of dynamic component in CSI values are proportional to the dynamic path length changes caused by the object movement. By measuring and analyzing the phase

978-1-5386-4281-8/18/$31.00 ©2018 IEEE

Table I
COMPARISON OF DIFFERENT WIFI-BASED SYSTEMS

System	Object	Granularity	Range	TX&RX
WiDraw [8]	Hand	5 cm	0.6 m	27
Widar [9]	Human body	25 cm	$0.8 \sim 3.2$ m	3
Wikey [1]	Gesture	Recognition	4 m	5
WiFinger [2]	Gesture	Recognition	$1 \sim 4$m	5
Wigesture [7]	Gesture	Recognition	$\geq$ 2m	≥ 2
WiTrace	**Gesture**	**2.09 cm**	$\geq$ **2 m**	**3**

changes, WiTrace achieves an average distance error of 3.75 cm when pushing hand for 30 cm in the range of 2 m using omnidirectional antennas.

The second challenge is to separate the phase changes caused by the moving hands from CSI values caused by other environments. The Signal-to-Noise Ratio (SNR), which represents the ratio of the reflecting power of target objects and other static objects, attenuates at long distance. As a result, the phase changes caused by the moving hands can be easily contaminated by other ambient interference, which means it is challenging to extract the phase changes from mixture signals. To address this challenge, we apply a heuristic algorithm, *i.e.* Extracting Static Component (ESC) which lies in its robustness to the ambient interference. For In-phase or Quadrature components of CSI, we first find the nearby local maxima and minima using empirical threshold. To wipe out those noisy extreme points, we set temporal threshold that is determined by the maximal Doppler frequency.

The third challenge is to estimate the initial position of hand in 2D space. Although we can precisely measure the distance changes of hand movement, it is difficult to locate the absolute position of the hand directly without the initial hand location. To address this challenge, we utilize the fact that the trajectory would be different when tracing the hand with the same path length change for different initial positions. We use the result of two preamble gestures as the fingerprints of different initial positions and combine two directions to estimate the initial position. Our approach achieves estimated accuracy of the initial position 3.91 cm on average.

E. Summary of Experimental Results

We implemented WiTrace using USRP transceivers. The emitted signal is 802.11g WiFi signal with central frequency 2.4 GHz. Our experimental results show that our approach achieves estimated accuracy of the initial hand position 3.91 cm on average, and tracks the hand movement with mean accuracy of 1.46 cm for 1D tracking and 2.09 cm for 2D tracking, respectively. The result also shows that WiTrace reaches overall mean direction error of 7.32 degrees across five different directions in 2D space case.

II. CSI PHASE MODEL

In this section, we describe the theoretical model of Channel State Information (CSI) regarding dynamic gesture movement as a background introduction. Specifically, CSI estimates the channel properties of a communication link, which is described by channel frequency response (CFR) for carrier frequency f [18]. The transient CSI value at time t with carrier frequency f, which is denoted by $H(f,t)$, can be derived by

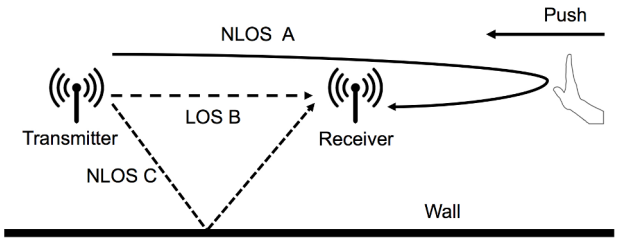


Figure 1. Illustration of multiple paths

$H(f,t) = Y(f,t)/X(f,t)$, where $X(f,t)$ and $Y(f,t)$ are the Fourier expressions of transmitted and received signals in frequency domain, respectively. As shown in Figure 1, there are multiple paths where signals propagate from the transmitter to the receiver. As a result, the CSI of the wireless channel at time t and frequency f is the superimposition response of all paths [19]:

$$H(f,t) = \left(\sum_{i=1}^{K} \alpha_i(t) e^{j(2\pi f d_i(t)/c + \phi_i)}\right) e^{j\psi(f,t)} \quad (1)$$

where K is the total number of paths, $\alpha_i(t)$ is the attenuation coefficient, $d_i(t)$ is the length of path i, c is the speed of wireless signal, and ϕ_i is the initial phase caused by time delay of the imperfect hardware. Additionally, traditional CSI measurements typically have phase shift $\psi(f,t)$, which is caused by residual frequency offset due to non-synchronized-locks between transceiver pair. In order to rule out the phase errors, we use an external clock [20] to connect the transmitter and the receiver in our system.

As shown in Figure 1, all of the paths can be divided into static paths, *e.g.* wall and LoS path, and dynamic paths *e.g.* hand. For static path i, the length of path d_i can be considered as a fixed distance during a short period. As a result, Eq. (1) can be rewritten as:

$$H(f,t) = H_s(f) + \sum_{i \in P_d} \alpha_i(t) e^{j(2\pi d_i(t)/\lambda_f + \phi_i)} \quad (2)$$

where $H_s(f)$ is the sum of CSI for the static paths, P_d is the set for the dynamic paths, and $\lambda_f = c/f$ is the wavelength for frequency f.

Suppose we can derive the phase change of path i, *i.e.*, $\Delta\varphi_i$, where the phase information φ_i is $\varphi_i = 2\pi d_i(t)/\lambda_f + \phi_i$. Thus, the length change of dynamic path i is given by:

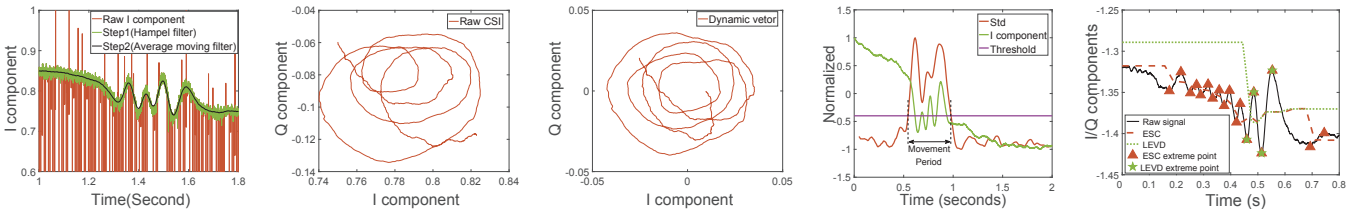
$$\Delta d_i = \frac{\Delta\varphi_i \lambda_f}{2\pi} \quad (3)$$

where $\Delta\varphi_i$ is the phase change of path i.

Finally, our goal is to measure the phase changes of the dynamic path caused by hand movement, and thereby determine the length change of dynamic path to track hand in the air.

III. CSI PHASE BASED DISTANCE MEASUREMENT

In this section, we propose a method to measure hand movement. Our measurement method contains five steps, as shown in Figure 3. First, we apply the Hampel filter to remove the noise of CSI signal. Second, we verify the CSI phase model by illustrating the phase of CSI signal in two dimensions. Third, we use variance of CSI amplitude to detect the start of the movement. Fourth, we propose a heuristic algorithm to



(a) CSI preprocessing (b) I/Q trace of raw CSI (c) I/Q trace of dynamic vector (d) Variance of I/Q component (e) Static estimation based on LEVD and ESC

Figure 2. 1D tracking measurement

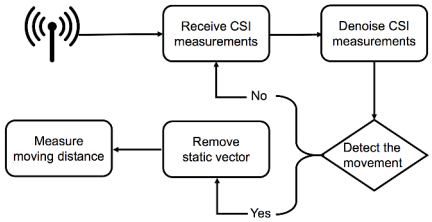


Figure 3. Processing flow for 1D

remove the static vector for CSI signal. At last, we transform the phase change of CSI signal to the movement distance.

A. CSI Signal Preprocessing

As shown in Figure 2(a), raw collected CSI signals as shown in red curves demonstrate extraordinary jitters as they contain various types of noise [21]. On one hand, there are many outliers in CSI signal, which is mainly caused by wireless interference. On the other hand, system hardware may generate high frequency noise [22]. As a result, we need to preprocess the CSI signal to remove its noise for better system performance. First, we apply the Hampel filter [23] to filter out the I/Q component outliers which have significant different values from others. The green curve in Figure 2(a) shows the results after the processing of Hampel filter. Second, we utilize the moving average low-pass filter to further remove high frequency noise. The black curve in Figure 2(a) shows the result of the CSI signal after signal preprocessing. Then, we obtain CSI measurements by integrating I component and Q component from all of the subcarriers.

B. CSI Phase Model Verification

To better understand CSI phase model, we use the real world CSI measurement to illustrate the CSI phase model. Figure 2(b) presents how the CSI phase changes by using the CSI measurement after preprocessing. During a short time period from 1.25 to 1.7 seconds, a user pushes his hand for 28 cm towards the receiveing end. We find that CSI rotates clockwise for $\Delta d/\lambda = 28 \times 2/12.5 \approx 4.5$ turns. We further observe that the static vector which corresponds to the static path is not absolutely constant during this period. This is mainly due to other slow changes around ambient environment during the hand movement. Meanwhile, the CSI amplitude is not stable during the period. This is owing to the increasing strength of reflected signal. As a result, we require to remove the static vector and extract the dynamic vector, which corresponds to the dynamic path, to get the phase change of dynamic vector as shown in Figure 2(c).

C. Movement Detection

Before measuring the movement, our system needs to detect the start of the movement. When user keeps static, the amplitude of CSI is stable except small fluctuation caused by ambient noise. Meanwhile, once the user begins to move his hand, the amplitude of CSI experiences large fluctuations because of phase change. We apply the sliding window to compute the variance of the amplitude continuously. In Figure 2(d), Std represents the standard deviation of each short period, and I component means the In-phase component of CSI values. To make it more convenient, both Std and I component are normalized to $[-1, 1]$. As shown in Figure 2(d), the variance in static period is much smaller than the variance in dynamic period. So the movement period can be easily detected by using experienced threshold. However, there still may exist some abnormal variances due to multipath effect for one frequency. These multipaths are mainly caused by movement of other body parts. As a result, we combine the results of all the subcarriers by using mathematical expectation to migrate the effect of multipath. Then, we use a predefined empirical threshold to detect the beginning and end of the movement.

D. Static Vector Elimination

In reality, it is challenging to remove static vector from the CSI measurement. On one hand, the static vector mainly caused by static reflectors, *e.g.*, Path B and Path C as shown in Figure 1, is much stronger than dynamic vector caused by hand, *e.g.* Path A. On the other hand, static vector may change slowly with the moving of hand due to blocking other reflectors and the slow movement of other body parts (*e.g.* arm). Additionally, even though dynamic vector of hand dominates the variation of CSI, SNR will degrade with distance between hand and receiving end.

There are some existing works to separate static vector from dynamic vector of hand. The mmWave radio system, mTrack, [11] uses Dual-Differential Background Removal (DDBR) and Phase Counting and Reconstruction (PCR) to remove the static vector. Compared to the 60 GHz signal, CSI signal is more susceptible to ambient noise. However, DDBR requires low surrounding noise and can hardly detect slow movement. PCR needs strong periodicity in the CSI signal. Both of these methods are, therefore, not suitable for our situation. LLAP [14] based on ultrasound applies a heuristic algorithm called Local Extreme Value Detection (LEVD) based on Empirical Mode Decomposition (EMD) algorithm [24] to estimate the static vector. It isolates the static vector by detecting whether

Algorithm 1: Extracting Static Component Algorithm

Input: CSI signal of real and imaginary part $I(t)$ and $Q(t)$. CSI deviation of real and imaginary part d_s^I and d_s^Q for previous static period.
Output: Estimated static vector $S(t)$

Initialize extrema of real and imaginary part: $E^I(t_i^I)$, $E^Q(t_i^Q)$, where t_i^I, t_i^Q are the timestamps of extrema;
for *each time* t **do**
  if *detect the movement* **then**
    /*Find extreme point of $I(t)$ */
    if $I(t)$ *is a local extrema and* $|I(t) - E^I(t_i^I)| \geq d_s^I$ **then**
      Take STFT of $I(t)$ to get the maximum doppler frequency f_{max}^I;
      if $t - t_i^I \geq 1/(2f_{max}^I) - \mu$ **then**
        $i \leftarrow i + 1$;
        $t_i^I \leftarrow t$;
        $E^I(t_i^I) \leftarrow I(f, t)$;
      $I_s(t) \leftarrow (E^I(t_i^I) + E^I(t_{i-1}^I))/2$;
    /*Find extreme point of $Q(t)$ */
    if $Q(t)$ *is a local extrema and* $|Q(t) - E^Q(t_j^Q)| \geq d_s^Q$ **then**
      Take STFT of $Q(t)$ to get the maximum doppler frequency f_{max}^Q;
      if $t - t_i^Q \geq 1/(2f_{max}^Q) - \mu$ **then**
        $j \leftarrow j + 1$;
        $t_i^Q \leftarrow t$;
        $E^Q(t_i) \leftarrow Q(t)$;
      $Q_s(t) \leftarrow (E^Q(t_i^Q) + E^Q(t_{i-1}^Q))/2$;
    /*Update the static vector*/
    $S(t) \leftarrow I_s(t) + jQ_s(t)$;
return $S(t)$;

the gap between alternate local maximum and minimum points is larger than an empirical threshold Thr. Here Thr is set as three times of the standard deviation of the baseband signal in a static environment. However, for CSI signal, the static vector is always contaminated by surrounding noise in the whole process, thus it is difficult to reliably detect the local maximum and minimum points by threshold Thr. For example, most maximum and minimum points in Figure 2(e) are failed to be detected by LEVD. As a result, we propose the so-called Extracting Static Component (ESC) method as shown in Algorithm 1 to estimate static vector. On one hand, instead of using threshold that is three times of the standard deviation, we use empirical threshold called Thr_m, which is far below the previous threshold to avoid neglecting all local extreme points. However, this operation may include some noisy points leading to incorrect static components. To remove noisy extreme points in the environment, we set temporal threshold related to the frequency shift of signals caused by gesture movement, say T_d, to $1/(2f_{d_{max}}) - \mu$, where $f_{d_{max}}$ is the largest Doppler frequency shift for each short period and μ is a small positive constant. We apply Short Time Fourier Transform (STFT) method on CSI measurements to derive the instantaneous Doppler frequency shift. The duration between any two adjacent extreme points should be more than T_d, which is a little larger than half of the shortest period. As shown in Figure 2(e), ESC improves the accuracy to detect the movement of hand and avoiding the small noise induced by ambient environment within the first 0.15 seconds.

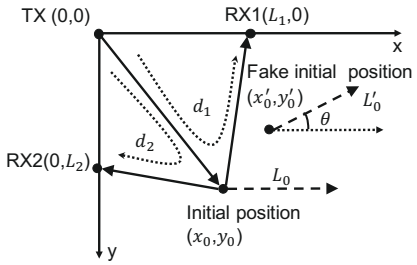


Figure 4. 2D geometric scenery

E. Distance Measurement

After detecting the movement and removing static vector, the phase of dynamic vector changes linearly with path length change, according to Eq. (3). As Figure 1 shows, since transmitter/receiver and hand are set on the same line, the real movement distance of hand is half of the path length, *e.g.*, $\Delta d = \frac{d_{t_i} - d_{t_j}}{2}$ when a user pushes his hand from moment t_i to t_j. Although we have mitigated the effect of static multipath by removing the static vector, there still remains some dynamic multipath effect when hands move. We utilize the fact that different subcarriers have different frequencies. Meanwhile, when there is no multipath effect, the measured distance changes should be the same for all subcarriers, while the phase changes of different subcarriers are different. As a result, we combine results of different subcarriers by using linear regression to mitigate the multipath effect further.

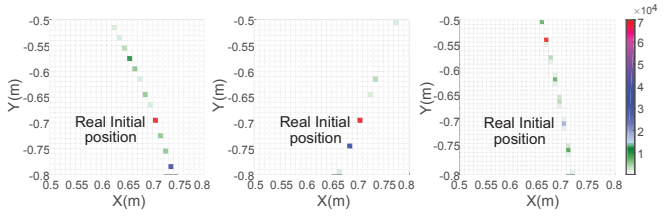
IV. 2D TRACKING

In this section, we present our 2D tracking algorithm based on the distance measurements in Section II. First, we use the preamble gestures to estimate the initial position. Second, we model 2D gesture tracking by using one transmitter and two receivers. At last, we present a Kalman filter based algorithm to correct the trajectory.

A. Initial Position Estimation

As we will see, the trajectory of hand in 2D space is not only determined by the path length change of the movement but also by its initial position. The incorrect initial position of hand will severely affect the tracking result. As shown in Figure 4, a user pushes his hand from (x_0, y_0) along the X-axis and the movement distance is L_0. For the same path length change, if the initial position is (x_0', y_0'), the movement distance would be changed to L_0', and the trajectory direction would have deviation of θ.

We have evaluated several methods to estimate the initial position in 2D space. For example, mTrack [11] based on 60 GHz technology uses a discrete beam scanning mechanism to pinpoint the object's initial localization. They use two RX antennas of beamwidth 4.5° to steer the beam. The intersection of two receiving beams is used to estimate the initial position. However, this method can only operate in a small range and is dependent on the beamwidth of directional antenna. Another ultrasonic work, LLAP, uses Inverse Discrete Fourier Transform to process CFR signals for all subcarriers to get



(a) Horizontal simulation (b) Vertical simulation (c) Real horizontal experiment

Figure 5. Initial position

the objects's absolute position. However, the calibration of the algorithm requires the exact distance between the reflector and receiver to compensate the initial phase ϕ_i in Eq. (1). Additionally, the bandwidth B of signals needs to be large enough such that the resolution of absolute path length (*i.e.*, c/B) is accurate. Since the bandwidth of 802.11g WiFi Orthogonal Frequency Division Multiplexing (OFDM) signal is only 20 MHz, the resolution of using IDFT is $\frac{c}{B} = \frac{3\times10^8}{20\times10^6} = 15$ m. It is obviously not suitable for our case. We estimate the initial position of hand by performing preamble gestures before the tracking. Our algorithm is inspired by the fact that the trajectory would be different for various candidate initial positions with the same path length change. First, the user is asked to push hand along X-axis and Y-axis, respectively. Then, we use the result of these two preamble gestures as the fingerprint of different initial positions. At last, we find the least deviation position to estimate the optimal initial position.

To better understand this method, we simulate the whole process of our method. We suppose that a user moves his hand along X-axis for 30 cm. Then, we use the measured distance change to determine the initial position with maximum probability. Generally, the gesture trajectory (x, y) related to candidate initial position (x', y') can be determined by solving the following two equations:

$$\begin{cases} \sqrt{x'^2+y'^2}+\sqrt{(x'-L_1)^2+y'^2} \\ =\sqrt{x^2+y^2}+\sqrt{(x-L_1)^2+y^2}+\Delta d_1 \\ \\ \sqrt{x'^2+y'^2}+\sqrt{x'^2+(y'-L_2)^2} \\ =\sqrt{x^2+y^2}+\sqrt{x^2+(y-L_2)^2}+\Delta d_2 \end{cases} \tag{4}$$

where Δd_1 and Δd_2 are the path length changes corresponding to two receivers. Figure 5(a) shows the simulating result for horizontal movement. The brightness represents the trajectory deviation of Y-axis for various initial positions (*i.e.*, $\arg\max_{y'} \frac{1}{|y'-y|}$). Similar to the horizontal movement, the brightness represents the trajectory deviation of X-axis for the vertical movement (*i.e.*, $\arg\max_{x'} \frac{1}{|x'-x|}$), as shown in Figure 5(b). From Figure 5(a) and Figure 5(b), we find that the darkest red point is the real initial position in our simulation. However, in reality, the real initial position may not be the darkest red point, since the distance measurement may have some errors. As shown in Figure 5(c), the real initial position is the second darkest red point of all the candidate positions. Thus, we combine X-axis and Y-axis directions to locate the initial point more accurately. Algorithm 2 shows the algorithm for locating the initial point.

Algorithm 2: Estimating Initial Position Algorithm

Input: Vertical movement path length changes of two receivers: $\Delta d_1^v, \Delta d_2^v$, horizontal movement path length changes of two receivers: $\Delta d_1^h, \Delta d_2^h$

Output: Estimated position (x_0, y_0) in two-dimensional space

for *each grid* (x_i, y_i) **do**
- Set the grid (x_i, y_i) as the candidate initial position;
- /*Vertical movement calculation*/
- Calculate the tracking trajectory $(\hat{x_i^v}, \hat{y_i^v})$ for two receivers based on the initial position (x_i, y_i) and path change $\Delta d_1^v, \Delta d_2^v$;
- /*Horizontal movement calculation*/
- Calculate the tracking trajectory $(\hat{x_i^h}, \hat{y_i^h})$ for two receivers based on the initial position (x_i, y_i) and path change $\Delta d_1^h, \Delta d_2^h$;

Find N candidate positions (x_i^v, y_i^v) which have the top N smallest deviations $(|\hat{x_i^v} - x_i|)$ of X-axis;

Find M candidate positions (x_i^h, y_i^h) which have the top M smallest deviations $(|\hat{y_i^v} - y_i|)$ of Y-axis;

Calculate $M \times N$ distance matrix $z_{M\times N}$, where $z_{i,j} = \sqrt{(x_i^h - x_j^v)^2 + (y_i^h - y_j^v)^2}$;

Find the smallest element $z_{i_{min},j_{min}}$ in matrix $z_{M\times N}$; $(x_0, y_0) \leftarrow (\frac{(x^h_{i_{min}}+x^v_{j_{min}})}{2}, \frac{(y^h_{i_{min}}+y^v_{j_{min}})}{2})$;

return (x_0, y_0) ;

B. Successive 2D Tracking

For 2D tracking, we use one transmitter and two receivers to track hand in two-dimensional space. As shown in Figure 4, the transmitter is set at point $(0, 0)$, while two receivers are set at $(L_1, 0)$, $(0, L_2)$, respectively. Suppose that the lengths of the two paths from transmitter to two receivers in Figure 4 are d_1 and d_2, respectively. Therefore, the position of the hand is exactly the intersection of the two ellipses defined by the following two equations:

$$\begin{cases} \frac{4(x-L_1/2)^2}{d_1^2} + \frac{4y^2}{d_1^2-L_1^2} = 1 \\ \\ \frac{4x^2}{d_2^2-L_2^2} + \frac{4(y-L_2/2)^2}{d_2^2} = 1 \end{cases} \tag{5}$$

where (x, y) is the coordinate of the hand position. Given the estimated initial point, d_1 and d_2 are calculated through updated distance change measurements (*i.e.*, Δd_1 and Δd_2) in Section IV-A. By solving these two equations with the conditions of $x > 0$ and $y < 0$, the realtime position of hand can be located, and the next successive position will be derived from this iterative method.

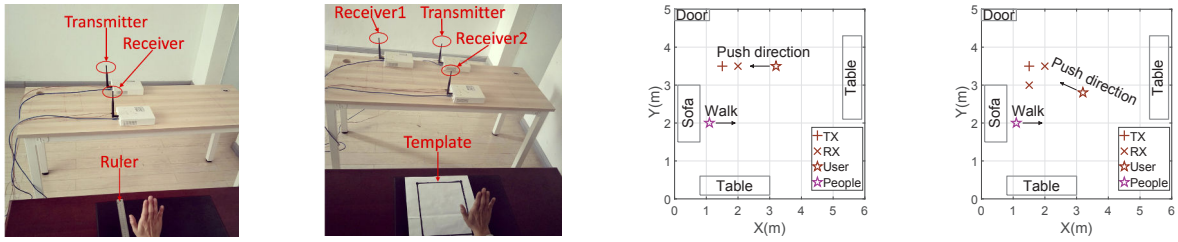
C. Trajectory Correction

The obtained coarse-grained trajectory of the hand still has large error due to persistent systematic noise. To filter out such noise and further improve the accuracy of the trajectory, we propose to use the Kalman filter based on a 2D Continous Wiener Process Acceleration (CWPA) model [25], which basically handles the case where the object's acceleration is perturbed by Gaussian noise, to correct the trajectory.

The state vector of 2D CWPA model is

$$\boldsymbol{s}_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k & \ddot{x}_k & \ddot{y}_k \end{bmatrix}^{\mathrm{T}} \tag{6}$$

where (x_k, y_k), $(\dot{x}_k, \dot{y}_k)$ and $(\ddot{x}_k, \ddot{y}_k)$ are the movement distance, velocity and acceleration of the hand at time k,



(a) 1D scenario (b) 2D scenario (c) 1D setup (d) 2D setup

Figure 6. Evaluation environment in laboratory

respectively. The Kalman filter model assumes the true state at time k is evolved from the state at $(k-1)$ according to

$$\boldsymbol{s}_k = \boldsymbol{A}\boldsymbol{s}_{k-1} + \boldsymbol{q}_k \tag{7}$$

where only the movement acceleration have the process noise, $\boldsymbol{q}_k \sim \mathcal{N}(0, \boldsymbol{Q})$, $\boldsymbol{Q}$ is the covariance matrix of the process noise. Based on the physical laws of motion, the transition matrix is

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & t & 0 & \frac{1}{2}t^2 & 0 \\ 0 & 1 & 0 & t & 0 & \frac{1}{2}t^2 \\ 0 & 0 & 1 & 0 & t & 0 \\ 0 & 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The measurement vector at time k in our system is

$$\boldsymbol{z}_k = [x_k \quad y_k]^{\mathrm{T}} \tag{9}$$

where (x_k, y_k) is the movement distance. As a result, the measurement $\boldsymbol{z}_k$ of the true state $\boldsymbol{s}_k$ is made according to

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{v}_k \tag{10}$$

where $\boldsymbol{v}_k$ is the measurement noise, $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R})$, $\boldsymbol{R}$ is the covariance of the measurement noise. The observation matrix is

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

We use the model above to follow the traditional steps of Kalman filter to correct trajectory immediately.

Note that other trajectory correction methods (*e.g.*, Roughness Penalty Smoothing and particle filter) have the similar performance (as shown Figure 9) to ours, they are not suitable for our system for their high computation cost and large processing delay.

V. Implementation and Evaluation

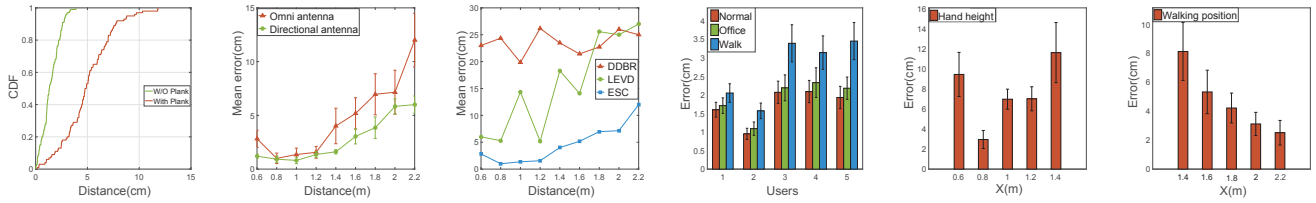
A. Implementation

We implemented WiTrace on the software radio platform-USRP-N210 hardware similar to another communication platform (*i.e.* NB-IoT terminal) [26]. The transmitting USRP with SBX board sends IEEE 802.11g OFDM frames [27] with bandwidth 20 MHz at 2.4 GHz [28]. There are 64 subcarriers in each transmitted frame, among which 48 subcarreiers are for data, 4 subcarriers are for pilot. Each receiver collects CSI measurements at a rate of 20 M samples per second using a laptop. Since each frame consists of 64 subcarriers, each receiver collects 64 CSI measurements per OFDM symbol. All subcarriers are modulated in Binary Phase Shift Keying. To reduce processing complexity, we downsample CSI streams for each subcarrier with downsample rate 100. Therefore, the sampling rate is reduced to $20\ \mathrm{M}/(64*100) = 3.125$ KHz for each subcarrier. The transmission and reception power is set to 20 dBm which is the same with the COTS WiFi NIC. For 1D tracking, our system uses one receiving USRP, as shown in Figure 6(a). For 2D tracking, two receiving USRPs are in perpendicular plane by default as shown in Figure 6(b). Both omnidirectional antennas and directional antennas (horizontal beam width $= 35°$ and vertical beam width $= 30°$) are used in our experiment. TX and RX antennas, together with user's hand, are at the height of 0.8 m. Our wireless transceiver system is synchronized by an external clock to avoid Carrier Frequency Offset and Sampling Frequency Offset, which changes the phase of CSI significantly. To synchronize two receivers, we send 1000 abnormal frames which has 10752 bytes per frame before tracking. The CSI measurements are processed using MATLAB offline.

B. Evaluation Metrics

Our experiments are conducted in a laboratory with the area of 5 m × 6 m. For 1D scenario, the transmitter and the receiver are set in a line, as shown in Figure 6(c). The distance between the transmitter and the receiver is 0.5 m.

We evaluate 1D tracking with omnidirectional antennas in terms of four metrics: (1) *Tracking accuracy*: the error between measured movement distance and ground truth movement distance measured by ruler along the movement path when the distance between receiver and user is set to 1.2 m. (2) *Tracking accuracy with different antennas*: the error between measured distance and the ground truth distance by using the omnidirectional antenna and directional antenna along the same movement path at different distances. (3) *Tracking accuracy with different algorithms*: the measurement error by using different algorithms to extract the phase changes along the same movement path at different distances. (4) *The robustness for different scenarios and users*: the measurement error for three different scenarios and five users. (5) *The effect of hand height and other people walking around on tracking accuracy*. For 2D scenario, the transmitter and the receivers are set as shown in Figure 6(d). We evaluated 2D tracking with omnidirectional antennas from three metrics: (1) *Initial position error*: the distribution of all estimated initial positions via preamble gesture. (2) *Tracking error*: the error between the measured trace and the standard template. (3) *The robustness for different scenarios and users for 2D*. (4) *The impact of different pushing directions on 2D tracking*.



(a) CDF for measurements error (b) Different antennas (c) Different algorithms (d) Different scenarios (e) Different hand heights (f) Different walking positions

Figure 7. 1D tracking error

C. Experimental Results

Tracking accuracy in 1D space: *WiTrace achieves average error of* 1.46 *cm when the hand moves for* 30 *cm at a distance of* 1.2 *m.* As shown in Figure 6(c), the initial position of volunteer is 1.2 m away from the receiver and the volunteer pushes hand for a distance of 30 cm towards the receiver. Figure 7(a) shows the Cumulative Distribution Function (CDF) of the distance error for 100 movements. The 80th percentile measurement error is 2.59 cm and the average error is 1.46 cm. Note that if we place a 5 cm thick plank (50 cm $\times$ 50 cm) between the transmitter and receiver, the average tracking error will slightly increase to 4.99 cm due to power loss during WiFi penetrating plank.

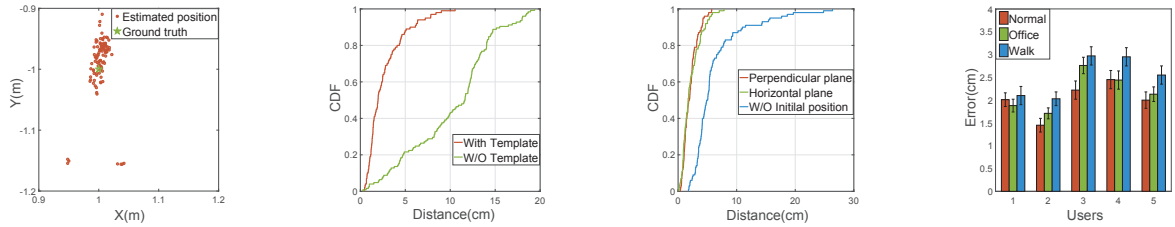
Tracking accuracy with different antennas in 1D space: *WiTrace achieves average error of* 3.75 *cm and* 2.51 *cm when using omnidirectional antennas and directional antennas in the range of* 2 *m, respectively.* Figure 7(b) shows the average error by using omnidirectional antennas and directional antennas when a volunteer is at different distances from the receiver. The results show that WiTrace achieves an average distance error of 3.75 cm and 2.51 cm when the hand moves for 30 cm over a distance of less than 2 m for omnidirectional antenna and directional antenna, respectively. The tracking errors for both omnidirectional antenna and directional antenna increase with the distance, due to the reducing SNR. Note that tracking error increases while user is quite close to the receiving end. This is because the reflection of limb and other dynamic body parts movement contaminate the dynamic vector by hand. The average accuracy of directional antenna outperforms 56% on average than omnidirectional antenna. This is due to the highly directional property of directional antenna, which leads to larger SNR than omnidirectional antennas at the same distance.

Tracking accuracy with different algorithms in 1D space: *By using ESC, WiTrace achieves highly improvement than other algorithms.* As shown in Figure 7(c), the average errors are 3.75 cm, 13.80 cm and 23 cm by using ESC, LEVD, and DDBR in the range of 2 m, respectively. Results show that the ESC algorithm outperforms the DDBR algorithm and the LEVD algorithm since the ESC algorithm is less susceptible to noise and robust for different distances.

The robustness for different scenarios and users in 1D space: *WiTrace is robust to background activities which are* 2 *m away from the receiver for different users.* To evaluate the robustness of WiTrace, we invited five users to push their hands towards receiver at a distance of 1.2 m while other volunteers were working or walking 2 m away from the receiver. The users indices 2 and 3 correspond to females while others are males ranging in age from 20 to 62. As shown in Figure 7(d), the standard deviation of different users is 0.86 cm, which shows WiTrace is robust for different users. Figure 7(d) shows the measurement errors for three different scenarios: "Normal" is a typical silent indoor scenario, "Office" represents the scenario that a person striking the keyboard near the user, and "Walk" represents the scenario that a person walk around 2 m away from the receiver. The average errors of "Normal", "Office", and "Walk" are 1.78 cm, 1.85 cm and 2.72 cm, respectively. WiTrace has slightly larger tracking errors for "Office" and "Walk" scenario than "Normal" scenario. This is mainly because micro striking action and walking around slightly affect the SNR of pushing.

The impact of hand height and people walking in 1D space: *WiTrace achieves average tracking error of* 6.46 *cm and* 3.80 *cm while pushing hand at the height from* 0.6 *m to* 1 *m and another people walking at the distance from* 1.6 *m to* 2.2 *m relative to the receiver, respectively.* As shown in Figure 7(e), the tracking error becomes larger while the hand height is further away from the height of the plane of transceiver (*i.e.*, 0.8 m). This is because that the practical path length caused by hand is larger than the pushing distance and the SNR is lower for longer paths. Similarly, the closer distance between another walking volunteer and the receiver will decrease the SNR of our system shown in Figure 7(f). The lower SNR will make it more difficult to detect movement of hand accurately, leading to larger tracking errors. The result above shows that WiTrace has high precision even if the user's hand is not at the same height with transceiver or another people walking around the user.

Estimated initial position error in 2D space: *WiTrace achieves average* 3.91 *cm estimated error with the template, and average* 10.18 *cm error without template in 2D space.* Figure 8(a) shows the estimated initial positions when users perform 100 pairs of vertical movements and horizontal movements along the template with the real initial position $(1, -1)$. Both vertical and horizontal movement distances range from 15 cm to 30 cm. Figure 8(b) show that 80th percentile estimated distance error is within 4.32 cm with template. This is mainly because the dynamic path measurements have slightly error, which leads to the error of estimated position according to our model. Additionally, we ask users to perform the same movement without the template. The result shows that the 80th estimating error is 14.05 cm, larger than the



(a) Initial position error in the 2D space

(b) CDF for estimated initial position

(c) CDF of tracking error for 2D

(d) Different scenorios

Figure 8. 2D measurement

previous result. This is because the movement of user's hand is more random than pushing along the template.

Tracking error in 2D space: *For 2D tracking, WiTrace achieves an average tracking error of* 2.09 *cm.* Figure 9 shows samples of three shapes' trajectories (*i.e.*, rectangle, triangle, and circle) drawn by WiTrace. We calculate each trajectory with time interval of 0.008 seconds for adjacent two points. The average time for all users to finish drawing rectangle, triangle, and circle are 2.8 seconds, 2.1 seconds, and 2.5 seconds, respectively. The initial position of user's hand is at the distance of 1.2 m with respect to the transmitter, and the drawing areas are around 30×30 cm. Figure 8(c) shows the CDF of the 2D tracking errors of 100 drawing movements in "Normal" scenario, which is defined as the distance of all points on the trace to the nearest points on the template. The 90th percentile measurement error is 3.95 cm and the average error is 2.09 cm comparing with the ground truth when the two receivers are in default perpendicular plane as shown in Figure 6(d). Note that if two receivers are in the horizontal plane, the average measurement error is 2.15 cm, similar to the perpendicular setup. It indicates that different setups have little impact on the 2D tracking accuracy. Moreover, without the initial position, the tracking performance will degrade severely. As shown in Figure 8(c), we assume that the initial position is 30 cm away from the practical location, the tracking error will increase to 6.23 cm on average. Additionally, Figure 9 also shows that Kalman filter method improves the tracking accuracy effectively.

The robustness for different scenarios and users for 2D: *WiTrace achieves the largest tracking error of* 2.98 *cm for different people and scenarios.* Figure 8(d) shows the mean error of estimated trace for each user across 50 drawings with three different scenarios. The average tracking errors for the "Normal", "Office", and "Walk" are 2.03 cm, 2.18 cm, and 2.52 cm, respectively. Thus, the tracking performance of our system is robust to interference caused by minor movement and drastic movement.

Impact of pushing direction in 2D space: *WiTrace reaches overall mean direction error of* 7.32 *degrees and mean tracking error of* 1.66 *cm at a distance of* 1.2 *m from the transmitter for pushing along a straight line.* As shown in Figure 4, users are invited to push hands in straight lines for 30 cm with different degrees related to the RX2. Figure 10 shows the CDF of distance error and direction error for five different directions. WiTrace achieves high resolution for different pushing directions, which means the pushing directions have slight impact on the tracking resolution.

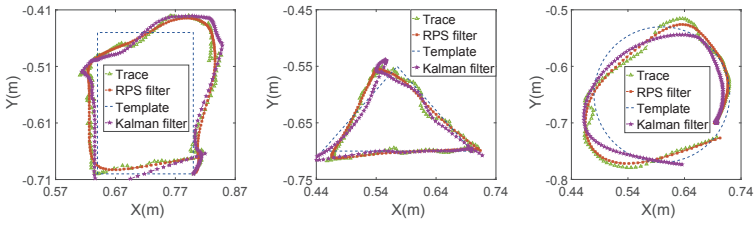
VI. Related Work

A. RF Based Gesture Recognition and Tracking

Recently, WiFi signals such as CSI, RSSI are used for gesture recognition [1]–[3]. WiKey proposes to use CSI dynamics to recognize keystrokes [1]. WiFinger uses CSI to recognize a set of eight gestures with accuracy of 93% [2]. WiGest uses the changes in WiFi RSSI through three wireless links to recognize a special set of gestures, and achieves a recognition accuracy of 96% [7]. Comparing with such recognizing systems, our scheme uses USRP to extract the CSI phase of WiFi signal to measure the quantifiable movement distance of hands. By using multiple antennas or receiving devices, RF based gesture tracking schemes are able to measure the movement distance and speed of hands [8], [11]–[13], [29], [30]. Unfortunately, these systems still have some drawbacks. some schemes require users to wear RF devices [29], [31], which makes them inconvenient to use. For example, the RF-IDraw scheme uses RFIDs attached on gloves to achieve tracking accuracy of 5.5 cm [29]. Some schemes have a limited tracking range, such as WiDraw [8] which has a working range of less than 2 feet. Furthermore, this scheme has to require multiple antennas with certain positions. Wideo leverages a software radio called WARP integrated into WiFi device to enable tracking accuracy of 7 cm [30]. However, they need to use antenna arrays and their mean localization error is 0.8 m. In contrast, WiTrace uses a single pair of WiFi devices to enable fine-grained accuracy of 3.91 cm for initial hand position and achieves average accuracy of 3.75 cm over a distance less than 2 m.

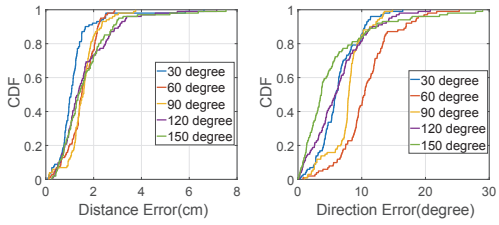
B. Non-RF Based Gesture Recognition and Tracking

Non-RF gesture recognition mainly includes vision based [4], [5], [32]–[35] and sound based [3], [14], [17], [36]. Vision based systems incur high computational cost and highly depends on the viewing angle and lighting conditions [4], [5], [33]–[35]. Sound-based system, LLAP [14], uses continuous wave signals to track hands and achieve accuracy of 3.5 mm and 4.6 mm for 1D and 2D tracking, respectively. FingerIO [17] proposed an OFDM based hand tracking system and achieves a hand location accuracy of 8 mm and allows 2D drawing in the air using COTS mobile devices. However, both of them only have a small tracking range which makes them unsuitable to serve as a remote control for home appliances. The key advantage of our scheme is that it is robust for different scenarios while tracking hands in a large range.



(a) Rectangle (b) Triangle (c) Circle

Figure 9. 2D trace



(a) CDF of distance error (b) CDF of direction error

Figure 10. 2D trace errors with various directions

VII. Limitations and Future Work

WiTrace establishes the feasibility of using WiFi signals to track gesture. However, our current implementation has some limitations. We currently calibrate the trajectory using our initial position estimating algorithm which requires user to take preamble gestures repeatedly. In this way, we determine the absolute position of hand avoiding accumulative error for long time tracking. Nevertheless, this method is inconvenient for user and we plan to explore the absolute phase of CSI which is related to the path length to refine the tracking results avoiding accumulative error. Second, we did not derive the accurate CSI phase from commodity WiFi devices. Since recent work shows that the accurate CSI phase can also be derived from commodity WiFi devices, such as Atheors AR9380 NICS [37], we plan to implement the system on the commercial WiFi NICs instead of software radio devices in future work.

VIII. Conclusions

In this paper, we made three key contributions. First, we utilize the phase changes of CSI to achieve high accuracy gesture tracking using WiFi signal. Second, to enable the 2D tracking, we propose a scheme based on two preamble gestures to measure the initial position of the hand in the 2D space. Third, we implement WiTrace on USRP and conducted comprehensive evaluation. Our experimental results show that WiTrace achieves cm-level tracking accuracy.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China under Grant 61502229, 61672276, 61472184 and 61321491, and the Jiangsu Innovation and Entrepreneurship (Shuangchuang) Program.

References

[1] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," in *Proc. ACM MobiCom*, 2015.
[2] S. Tan and J. Yang, "WiFinger: leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. ACM Mobihoc*, 2016.
[3] K. Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, and S. Patel, "Airlink: sharing files between multiple devices using in-air gestures," in *Proc. ACM UbiComp*, 2014.
[4] "Microsoft Kinect," http://www.microsoft.com/en-us/kinectforwindows/.
[5] "Leap Motion," https://www.leapmotion.com/.
[6] S. Nirjon, J. Gummeson, G. Dan, and K. H. Kim, "Typingring: A wearable ring platform for text input," in *Proc. ACM Mobisys*, 2015.
[7] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi based gesture recognition system," in *IEEE INFOCOM*, 2015.
[8] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "WiDraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proc. ACM MobiCom*, 2015.
[9] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi," in *Proc. ACM Mobihoc*, 2017.
[10] "Google project soli," https://www.google.com/atap/project-soli/.
[11] T. Wei and X. Zhang, "mTrack: High-precision passive tracking using millimeter wave radios," in *Proc. ACM MobiCom*, 2015.
[12] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. Usenix NSDI*, 2013.
[13] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person motion tracking via RF body reflections," in *Proc. Usenix NSDI*, 2015.
[14] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. ACM MobiCom*, 2016.
[15] Y. W. Huijie Chen, Fan Li, "Ecotrack: Acoustic device-free hand tracking on smart phones," in *Proc. IEEE INFOCOM*, 2017.
[16] H. Z. L. Q. Sangki Yun, Yi-chao Chen and W. Mao, "Strata: Fined-grained device-free tracking using acoustic signals," in *MobiSys*, 2017.
[17] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proc. ACM CHI*, 2016.
[18] "Enhancements for higher throughput," IEEE Standard 802.11n, 2009.
[19] J. G. Proakis, "Digital communications," *McGraw-Hill*, 1995.
[20] "OctoClock-G," https://www.ettus.com/product/details/OctoClock-G.
[21] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. ACM MobiCom*, 2015.
[22] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. ACM MobiCom*, 2015.
[23] L. Davies and U. Gather, "The identification of multiple outliers," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 782–792, 1993.
[24] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1998.
[25] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
[26] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, "Narrowband internet of things: Implementations and applications," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2309–2314, 2017.
[27] "IEEE 802.11g-2003," https://en.wikipedia.org/wiki/IEEE_802.11g-2003.
[28] IEEE computer society LAN MAN standards committee, "IEEE Std 802.11-1999," *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999.
[29] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: virtual touch screen in the air using RF signals," in *Proc. ACM SIGCOMM*, 2014.
[30] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using RF backscatter," in *Usenix NSDI*, 2015.
[31] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. ACM MobiCom*, 2014.
[32] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *MobiCom*, 2015.
[33] J. Song, G. Sörös, F. Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges, "In-air gestures around unmodified mobile devices," in *Proc. ACM UIST*, 2014.
[34] "Microsoft Hololens," http://www.microsoft.com/microsoft-hololens.
[35] X. A. Chen, J. Schwarz, C. Harrison, J. Mankoff, and S. E. Hudson, "Air+touch: interweaving touch & in-air gestures," in *ACM UIST*, 2014.
[36] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proc. ACM CHI*, 2012.
[37] Y. Zhuo, H. Zhu, H. Xue, and S. Chang, "Perceiving accurate CSI phases with commodity WiFi devices," in *Proc. IEEE INFOCOM*, 2017.