WiTrace: Centimeter-Level Passive Gesture Tracking Using OFDM signals

Lei Wang, Ke Sun, Haipeng Dai, *Member, IEEE,* Wei Wang, *Member, IEEE,*
Kang Huang, Alex X. Liu, *Senior Member, IEEE,* Xiaoyu Wang, and Qing Gu *Member, IEEE*

Abstract—Gesture tracking is a basic Human-Computer Interaction mechanism to control devices such as IoT and VR/AR devices. However, prior OFDM 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 OFDM 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 an average accuracy of 6.23 cm for initial position estimation, and achieves cm-level accuracy, with average tracking errors of 1.46 cm and 2.09 cm for 1D tracking and 2D tracking, respectively.

Index Terms—CSI, Gesture Tracking

1 Introduction

Gesture tracking is a basic Human-Computer Interaction mechanism to control not only electronic Internet of Things (IoT) devices but also VR/AR devices. In smart homes, gestures are recognized to change the channel of TVs or adjust the temperature of air conditioners. In VR/AR applications, users use gestures to interact with devices, such as typewriting in the air [2]. Gesture tracking can also be used for writing-in-the-air and gesture-based games. Recently, OFDM based WiFi signals are widely used for passive sensing for gesture movement as [3]–[5] due to its particular advantages. In comparison with vision based methods [6], [7], 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 [8], which is convenient and saves the extra cost of wearable devices. Prior WiFi based gesture recognition systems extract features from reflected signals for different gestures [4], [9] 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. Existing WiFi based tracking schemes include WiDraw [10], Widar [11], Widar 2.0 [12], and QGesture [13]. WiDraw uses Angle-Of-Arrival (AOA) measurements to achieve a tracking accuracy of 5 cm, which allows the user to draw in the air in densely deployed areas with more than 25 WiFi transmitters surrounding the user. Widar and Widar 2.0 are human tracking schemes which treat the human body as one single object with decimeter-level resolution. Our recent work, QGesture, uses phase information extracted from WiFi signals to track human hands. However, due to the phase noises, QGesture has a limited accuracy of 5.5 cm at a distance of 2 m and needs to know the initial position before performing 2D tracking. The dominant technologies above are shown in Table 1. Other Radio Frequency (RF) and acoustic signal based tracking schemes use localization technologies to track gestures. WiTrack [14], [15] proposes to use specially designed Frequency-Modulated Continuous-Wave (FMCW) radar with a high bandwidth of 1.79 GHz to track human movement behind the wall with a resolution of 11 cm to 20 cm, which needs special hardware. Similar to RF signal, although acoustic tracking schemes [16]–[19] have high accuracy, these systems cannot serve as remote control interface for home applications due to limited working range.

In this paper, we propose WiTrace, a WiFi OFDM 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 a carrier frequency of 2.4 GHz and a bandwidth of 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. Furthermore, we adopt the Kalman Filter (KF) based method to filter out noise of tracking.

- *L. Wang, K. Sun, H. Dai, W. Wang, K. Kang, X. Wang, Q. Gu are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: wangl@smail.nju.edu.cn; kesun@smail.nju.edu.cn; haipengdai@nju.edu.cn; ww@nju.edu.cn; hkwany520@gmail.com; mg1633074@smail.nju.edu.cn; guq@nju.edu.cn).*
- *Alex X. Liu is with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China, and also with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: alexliu@cse.msu.edu).*

A preliminary version of this work appeared in the proceeding of IEEE SECON 2018 [1]. Manuscript received April 19, 2005; revised August 26, 2015. Corresponding author: Wei Wang.

Table 1
Comparison of different WiFi-based systems

System	Object	Granularity	Range	TX&RX
WiDraw [10]	Hand	5 cm	0.6 m	27
QGesture [13]	Hand	5.5 cm	2 m	3
Widar [11]	Human body	25 cm	$0.8 \sim 3.2$ m	3
Widar 2.0 [12]	Human body	75 cm	8 m	2
Wikey [3]	Gesture	Recognition	4 m	5
WiFinger [4]	Gesture	Recognition	$1 \sim 4$m	5
Wigesture [9]	Gesture	Recognition	$\geq$ 2m	≥ 2
WiTrace	**Gesture**	**2.09 cm**	$\geq$ **3.5 m**	**3**

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 [10]. In contrast, we leverage the fact that the phase changes of dynamic component of CSI are proportional to the path length changes caused by the object movement. By measuring and analyzing the phase changes, WiTrace achieves an average distance error of 1.46 cm when pushing hand for 30 cm in the range of 1.2 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 (*i.e.* I) or Quadrature (*i.e.* Q) 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 movements, it is difficult to locate the absolute position of the hand directly without the initial hand location. Existing indoor localization based on WiFi signals [20]–[22] can be used for initial location estimation. However, these systems only get the coarse location at decimeter level of the human body, which is insufficient for gesture tracking. To address this challenge, we first estimate the coarse initial hand position based on the CSI phase difference of variant subcarriers caused by hand movement. This coarse initial position estimation can narrow down the candidate region for the following fine estimation step so that the computation complexity of the fine estimation can be significantly reduced. Then, we utilize the fact that the estimated trajectory would be different for different initial positions. We use the result of two preamble gestures as the fingerprints of different initial positions and combine two directions to refine the initial position estimation. Our approach achieves an average accuracy of 6.23 cm for initial position estimation.

We implemented WiTrace using USRP transceivers. Our experimental results show that our approach achieves estimated accuracy of the initial hand position 6.23 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.

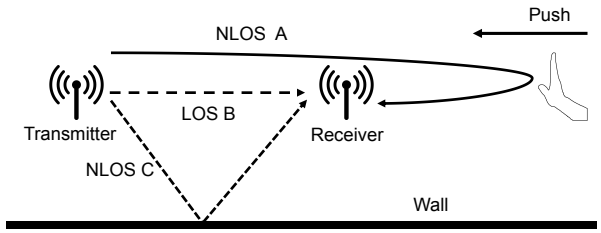


Figure 1. Illustration of multiple paths

2 CSI PHASE MODEL

In this section, we describe the theoretical model of Channel State Information (CSI) regarding dynamic gesture movement. Specifically, CSI estimates the channel properties of a communication link, which is described by channel frequency response (CFR) for k-th subcarrier frequency f_k [23]. As a result, CSI of the k-th subcarrier at time t is the superimposition response of all transmission paths [24]:

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

where K is the total number of paths, $\alpha_{i,t}^k$ is the attenuation coefficient of the k-th subcarrier, $d_i(t)$ is the length of path i, c is the speed of the wireless signal, and ϕ_i is the initial phase caused by time delay of the imperfect hardware. Additionally, traditional CSI measurements typically have a phase shift of $\psi(f_k, t)$, which is caused by residual frequency offset due to non-synchronized-clocks between transceiver pair. In order to rule out the phase errors, we use an external clock [25] 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.*, the wall and LoS path, and dynamic paths *e.g.*, the hand. For static path i, the length of path d_i can be considered as fixed during a short period. As a result, Eq. (1) can be rewritten as:

$$\overrightarrow{H}^k(t) = \overrightarrow{H}_{st}^k + \sum_{i\in P_d} \alpha_{i,t}^k e^{j(2\pi d_i(t)/\lambda_k+\phi_i)}, \quad (2)$$

where $\overrightarrow{H}_{st}^{f_k}$ is the sum of CSI for the static paths that are constant for a short duration, P_d is the set for the dynamic paths, and $\lambda_k = c/f_k$ is the wavelength for frequency f_k.

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

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

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.

3 CSI PHASE BASED DISTANCE MEASUREMENT

In this section, we propose a method to measure hand movement. Our measurement method contains four steps, as shown in Figure 2. First, we apply the Hampel filter to remove the noise of

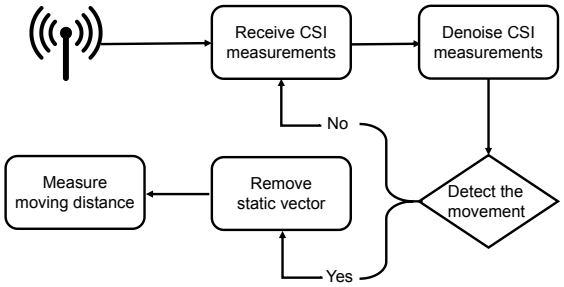


Figure 2. Processing flow for 1D tracking

CSI signal. After removing the noise, we can verify the CSI phase model by illustrating the CSI signal in two dimensions. Second, we use the variance of CSI amplitude to detect the start of the movement. Third, we propose a heuristic algorithm to remove the static vector from the CSI signal. At last, we transform the phase change of CSI signal to the movement distance.

3.1 CSI Signal Preprocessing

As shown in Figure 3(a), raw CSI signals (red curves) have large jitters as they contain various types of noise [26]. On one hand, there are many outliers in CSI signal, which are mostly caused by wireless interference. On the other hand, system hardware may generate high frequency noise [27]. As a result, we need to preprocess the I/Q components of the CSI signal to remove these noises. First, we apply the Hampel filter [28] to filter out the outliers that have are significantly different to others. The green curve in Figure 3(a) shows the results after applying the Hampel filter. Second, we utilize the moving average low-pass filter to further remove high frequency noise. The black curve in Figure 3(a) shows the result of the signal after filtering. We then get the complex valued CSI measurements using the I/Q components as the real/imaginary parts.

3.2 CSI Phase Model Verification

In Figure 3(b), we use a real world CSI measurement to illustrate how CSI phase changes in our CSI phase model. During a short time period from 1.25 to 1.7 seconds, a user pushes his hand for 28 cm towards the receiving end. As the two-way path length change is $\Delta d = 28 \times 2$ cm and the wavelength λ is 12.5 cm, we find that CSI rotates clockwise for $\Delta d/\lambda = 28 \times 2/12.5 \approx 4.5$ cycles. We further observe that the static vector which corresponds to the static path is not 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 need 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 3(c). Note that the centers of the traces of the dynamic vectors are moved to the origin so that we can directly measure the phase of them.

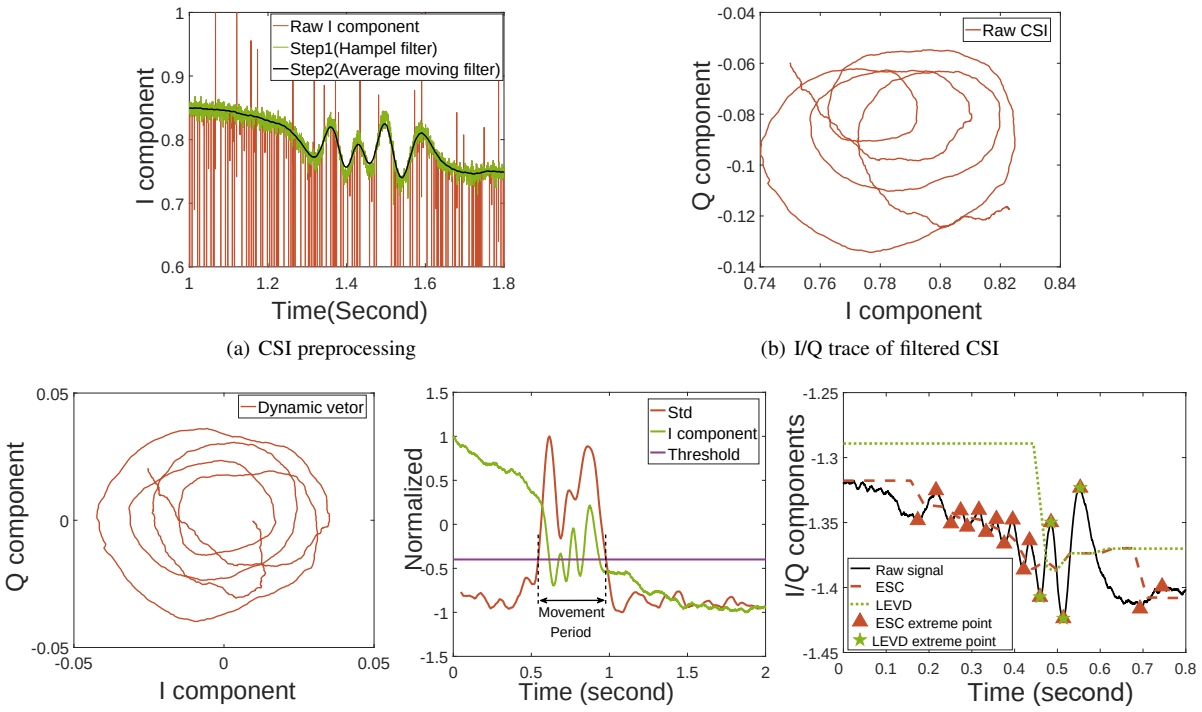
3.3 Movement Detection

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 for 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 a sliding window to compute the variance of the amplitude continuously. We choose to use the amplitude instead of the phase mainly due to the less computational cost of the amplitude as the unwrapping process for phase calculation may incur additional computation for rectifying the discontinuity in the phase. In Figure 3(d), *Std* represents the standard deviation of each short period, and I component means the In-phase component of CSI values. Both Std and I component are normalized to $[-1, 1]$ for a clear illustration. As shown in Figure 3(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 an empirical threshold. However, there may still 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 mitigate the effect of multipath. Then, we use a predefined empirical threshold to detect the beginning and end of the movement.

3.4 Static Vector Elimination

In reality, it is challenging to remove static vector from the CSI measurement. On one hand, the static vector that is mainly caused by static reflectors, *e.g.*, Path B and Path C as shown in Figure 1, is much stronger than the 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 of other reflectors and the slow movement of other body parts (*e.g.* the 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 algorithms that separate static vector from dynamic vector of hand. Dual-Differential Background Removal (DDBR) and Phase Counting and Reconstruction (PCR) are used in 60 GHz Radar systems, such as mTrack [29], to remove the static vector. However, both methods are not suitable for CSI signal due to the high noise-level of CSI and the lack of periodicity that is required by PCR. LLAP [16] based on ultrasound applies a heuristic algorithm called Local Extreme Value Detection (LEVD) based on Empirical Mode Decomposition (EMD) algorithm [30] to estimate the static vector. It isolates the static vector by detecting whether 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, 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 3(e) are failed to be detected by LEVD. As a result, we propose the 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 small 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 a temporal threshold related to the frequency shift of signals caused by gesture movement, 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



(a) CSI preprocessing (b) I/Q trace of filtered CSI

(c) I/Q trace of dynamic vector (d) Variance of I/Q component (e) Static vector estimation of LEVD and ESC

Figure 3. 1D tracking measurement

Algorithm 1: Extracting Static Component Algorithm

Input: CSI signal of real or imaginary part $X(t) = X^I(t)$ or $X^Q(t)$. CSI deviation of real or imaginary part $d_s = d_s^I$ or d_s^Q for previous static period.

Output: Estimated static vector $\overrightarrow{S}(t)$

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

frequency shift $f_{d_{max}}$. The duration between any two adjacent extreme points should be more than T_d, which is slightly larger than half of the shortest period. As shown in Figure 3(e), ESC improves the accuracy to detect the movement of hand and avoids the small noise induced by ambient environment within the first 0.15 seconds.

3.5 Distance Measurement

After movement detection and static vector elimination, 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 [16], which finds the best value of Δd that fits all phase changes obtained from different frequencies, to further mitigate the multipath effect.

4 2D TRACKING

In this section, we present our 2D tracking algorithm based on the distance measurements in Section 2. We first discuss the CSI measurement noise sources that make our initial position estimation results significantly deviate from ground truth and take methods to reduce the noises. Then, we apply phase difference over different subcarriers to coarsely estimate the initial hand position and propose a novel algorithm to refine the resolution subsequently. At last, we present a KF based algorithm to further refine the trajectory.

4.1 CSI Measurement Evaluation and Denoising

There are mainly three kinds of noises in CSI measurements that may give rise to large phase errors.

Carrier Frequency Offset (CFO): CFO occurs when there exists carrier frequency mismatch between the transmitter and receiver oscillators. Although the CFO is compensated by the

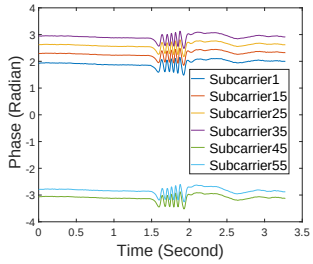


Figure 4. CSI phase changes of all subcarriers

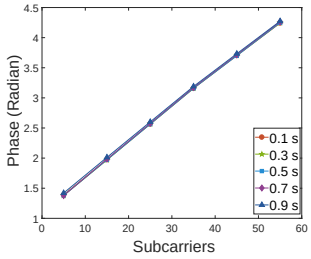


Figure 5. Raw CSI phase changes of various moment

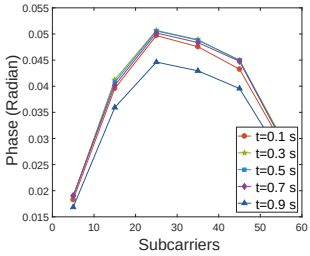


Figure 6. Denoised CSI phase of various moment

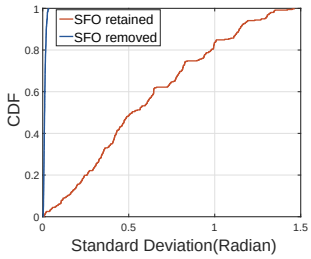


Figure 7. CDF of phase variance for all subcarriers W/O SFO

external clock in our system, the compensation is not perfect due to the hardware imperfection. The residual CFO will lead to the time-varying CSI values offset as $\overrightarrow{H}_c(f,t) = e^{j2\pi\Delta ft}\overrightarrow{H}(f,t)$, where Δf is the corresponding CFO. Fortunately, as shown in Figure 4, the frequency shift over a silent duration of 1.5 seconds is only 0.044 radians for all subcarriers, which is small, relative to the Doppler frequency caused by hand movements. Note that there are no any dynamic paths over this silent duration. For example, the Doppler frequency is at least 16 Hz when the velocity of a pushing hand is $1\ m/s$. This indicates that the impact of CFO can be ignored since it is small comparing to the impact of hand movements.

Sampling Frequency Offset (SFO) and Packet Detection Delay (PDD): In addition to CFO, two other noise sources (*i.e.* SFO and PDD) [13], [27] have the similar effect on the CSI measurements. As shown in Figure 5, the slopes of linear phase shift ϕ are almost the same on the different frames during the same measurement. However, the slope changes randomly across different measurements when the equipments restarts due to the randomness in the PLL locking process of the clock. Thus, we compensate the SFO/PDD offset whenever the system restarts. We use linear regression to remove the SFO/PDD offset [13]. Figure 6 shows the CSI phase before and after correction. We observe that the CSI phase is more consistent over different subcarriers after SFO/PDD correction. The largest phase difference for any two subcarriers is smaller than 0.03 radians, and there is only a small residual difference of less than 0.01 radians after phase correction. In case that the system restarts, CSI phase may have inconsistent initial phases. To demonstrate this, we collect 150 CSI samples across different measurements to compute the standard deviation of CSI phase over all subcarriers. Figure 7 shows the standard deviation of phase averaged over different subcarriers reduced from 0.61 radians before removing the CFO to 0.01 radians after removing SFO.

CSI variance: Although we have applied low-pass filter to remove high frequency noises at the denoising step, there is still CSI variance even in silent period. For example, the standard deviation of CSI phase for time period $0 \sim 1.5$ s as shown in Figure 4 is 0.013 radians. This variance will degrade the initial hand position estimation performance significantly. To remove CSI noises efficiently, we use three cascading low-pass filters to constitute a new cascading filter as $h'(t) = h(t) * h(t) * h(t)$, where $h(t)$ is the impulse response for one low-pass filter and $*$ represents convolution. Figure 8 shows the frequency response of the cascading filter comparing with the previous average moving filter. In Figure 9, we observe that high frequency noises are attenuated significantly by the filter so that the SNR is improved, where SNR here is denoted by the ratio of normal hand movements to the high frequency noises power on I/Q component. To demonstrate the effect of the filter, we collect 150 CSI samples with a duration of 96 ms. Then, we compute the standard deviation of CSI phase over the 96 ms segment for each subcarrier. Figure 10 shows that the average phase deviation significantly reduced from 0.032 radians before filtering to 0.025 radians after filtering. The denoising process is necessary for the coarse estimation in Section 4.2, since the performance of coarse estimation would degrade severely without denoising.

4.2 Coarse Initial Position Estimation

After removing CSI noises, we use the phase difference between subcarriers to coarsely estimate the initial position of hand. The coarse initial position estimation can narrow down the candidate region, which greatly reduces the complexity of the later refinement algorithm. As shown in Eq. (2) , CSI of one pair of transceiver is the superimposition of all paths. Suppose that there is only one dynamic path (*i.e.*, hand movement) in our system. Thus, the CSI vector of all paths on k-th subcarrier at time t can be described as:

$$\overrightarrow{H}^k(t) = \overrightarrow{H}^k_{st} + \alpha^k_{hd}(t)e^{j(2\pi d_{hd}(t)/\lambda_k + \phi_{hd})} \quad (4)$$

where H^k_{st}, $\alpha^k_{hd}(t)$, $d_{hd}(t)$, and ϕ_{hd} are the constant static vector, the attenuation coefficient, path length and initial phase for hand reflection, respectively. We can use $\overrightarrow{H}^k_{hd}(t) = \alpha^k_{hd}(t)e^{j(2\pi d_{hd}(t)/\lambda_k+\phi_{hd})}$ to represent the dynamic vector. As shown in Figure 11, the dynamic vector rotates due to the phase shift caused by hand movements. Furthermore, the phase difference of two subcarriers due to hand movement at time t can be described as:

$$\begin{aligned}\Delta p_{k_1,k_2}(t) &= arg(\overrightarrow{H}^{k_2}_{hd}(t)) - arg(\overrightarrow{H}^{k_1}_{hd}(t)) \\ &= 2\pi d_{hd}(t)/\lambda_{k_2} - 2\pi d_{hd}(t)/\lambda_{k_1}\end{aligned} \quad (5)$$

where we use k_1, k_2 to represent two different subcarriers and $arg(\overrightarrow{H}^k_{hd}(t))$ represents the the phase of dynamic vector corresponding to hand movement on subcarrier k. Once we know $\Delta p_{k_1,k_2}(t)$, the path length can be calculated as:

$$d_{hd}(t) = \frac{\Delta p_{k_1,k_2}(t)\lambda_{k_1}\lambda_{k_2}}{2\pi(\lambda_{k_1} - \lambda_{k_2})}. \quad (6)$$

We use the tangent vector corresponding to dynamic vector $\overrightarrow{H}^k_{hd}(t)$, denoted as $\overrightarrow{H}^k_{tg}(t)$, to estimate $\Delta p_{k_1,k_2}$. The phase change of the dynamic vector is equivalent to phase change in its tangent vector:

$$\begin{aligned}\Delta p_{k_1,k_2}(t) &= [arg(\overrightarrow{H}^{k_2}_{tg}(t)) - \pi/2] - [arg(\overrightarrow{H}^{k_1}_{tg}(t)) - \pi/2] \\ &= arg(\overrightarrow{H}^{k_2}_{tg}(t)) - arg(\overrightarrow{H}^{k_1}_{tg}(t)).\end{aligned} \quad (7)$$

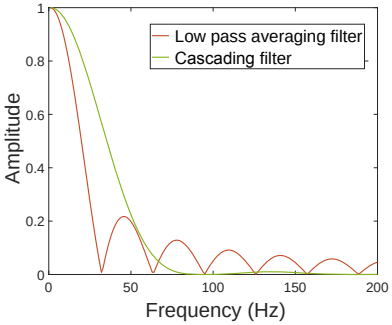


Figure 8. Frequency response of filter

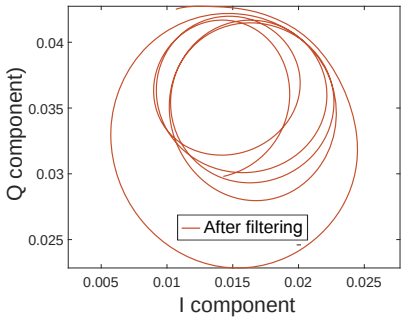


Figure 9. I/Q trace of CSI after filtering

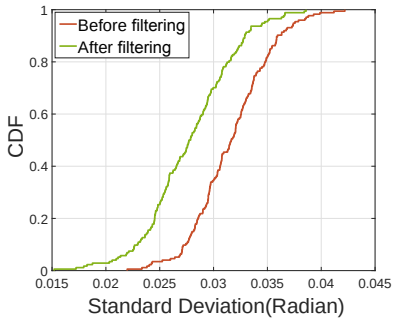


Figure 10. CDF of phase variance

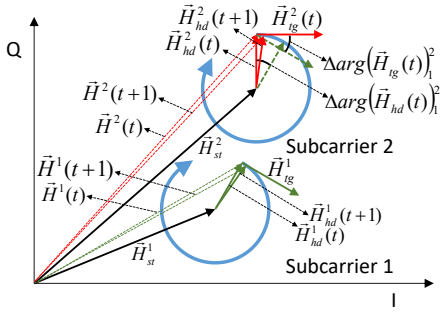


Figure 11. CSI vector changes

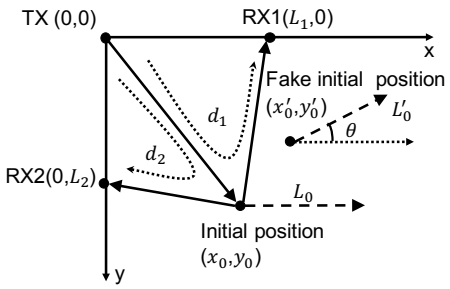


Figure 12. 2D geometric scenery

According to the definition of tangent vector, the tangent vector of dynamic vector $\overrightarrow{H}^k_{hd}(t)$ can be approximated as:

$$arg(\overrightarrow{H}^k_{tg}(t)) \approx arg(\overrightarrow{H}^k_{hd}(t+1) - \overrightarrow{H}^k_{hd}(t)) \quad (8)$$

where time $t+1$ is the next moment following time t. Since the static vector is constant for the short duration, Eq. (8) can be rewritten as:

$$\begin{aligned} arg(\overrightarrow{H}^k_{tg}(t)) &\approx arg([\overrightarrow{H}^k_{hd}(t+1) + \overrightarrow{H}^k_{st}] - [\overrightarrow{H}^k_{hd}(t) + \overrightarrow{H}^k_{st}]) \\ &\approx arg(\overrightarrow{H}^k(t+1) - \overrightarrow{H}^k(t)). \end{aligned} \quad (9)$$

Based on above equations, we derive the phase difference of two subcarriers due to hand movement at time t as:

$$\begin{aligned} \Delta p_{k_1,k_2}(t) \; = \; & arg(\overrightarrow{H}^{k_2}(t+1) - \overrightarrow{H}^{k_2}(t)) \\ & - arg(\overrightarrow{H}^{k_1}(t+1) - \overrightarrow{H}^{k_1}(t)). \end{aligned} \quad (10)$$

Combining Eq. (6) and Eq. (10), we directly derive the path length of hand $d_{hd}(t)$ corresponding to one pair of transceiver at time t.

To estimate initial hand position in 2D, we use one transmitter and two receivers in two-dimensional space. As shown in Figure 12, the transmitter is set at point $(0,0)$, while two receivers are set at $(L_1,0)$ and $(0,L_2)$, respectively. Suppose that the lengths of the two paths from transmitter to two receivers in Figure 12 are d_1 and d_2, respectively. Therefore, the position of the hand is the intersection of the two ellipses defined by the following two equations:

$$\begin{cases} \frac{4(x-L_1/2)^2}{d_1^2} + \frac{4y^2}{d_1^2-L_1^2} = 1 \\ \\ \frac{4x^2}{d_2^2-L_2^2} + \frac{4(y-L_2/2)^2}{d_2^2} = 1 \end{cases} \quad (11)$$

where (x,y) is the coordinate of the initial hand position.

The coarse initial position estimation still has two problems to be considered. Firstly, which two subcarriers should be chosen to compute phase difference in Eq.(5). On one hand, the frequency difference of these two subcarriers should not be too large. For example, if the frequency difference is 20 MHz, the maximal path length should be limited to 8.35 m to make sure that the phase difference does not exceed 2π. On the other hand, if the frequency difference is only 1 MHz, the estimated phase difference corresponding to the path length of 2 m will be only 0.02 radians, which can be easily contaminated by noise. In our system, we choose two subcarriers which have frequency difference around 10 MHz. Secondly, we need to choose appropriate time instance t to estimate the phase difference. We can only measure the dynamical vector when the hand is moving. However, to serve as an initial position, we need to measure the phase difference as soon as the hand moves. Thus, we perform phase difference estimation as soon as our system detects the movement as expressed in Section 3.3.

We use three experiments to evaluate coarse initial position estimation algorithm. Firstly, we ask the volunteer to move his/her hand at different distances to the transmitter. Figure 13 shows that the average estimation error is less than 0.31 m when the volunteer is at a distance of less than 2.5 m. The estimation error increases with the distance due to the weaker reflection signal at a longer distance. Secondly, we collect 150 CSI samples at a distance of 1.2 m to evaluate the estimation performance with/without the denoising algorithm. Figure 14 shows that the average estimation error with our denoising algorithm (*i.e.*, "Normal") is 0.28 m and the average error without filtering is 0.38 m. If we don't remove the SFO, the estimation error could be as large as 4.91 m. Such large random noise is due to the SFO introduced by the receiver clock. Thirdly, to evaluate the impact of different scenarios and users, we recruit five users to move their hands at a distance of 1.2 m while other volunteers were performing office tasks (*i.e.*, typing) or walking at 2.5 m away from the transmitter. Figure 15 shows

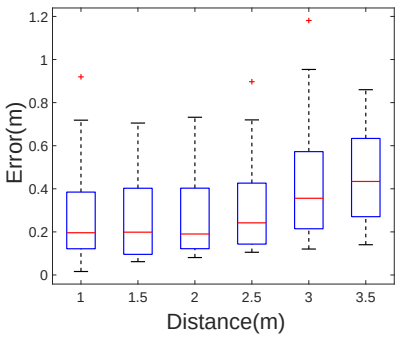


Figure 13. Coarse estimation error for different distances

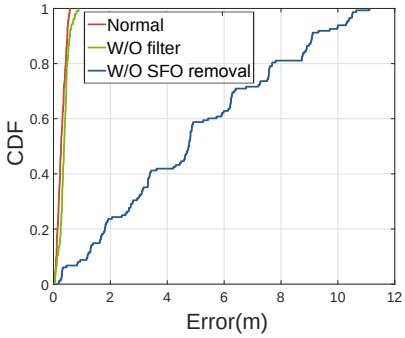


Figure 14. Coarse estimation error with and without denoising algorithm

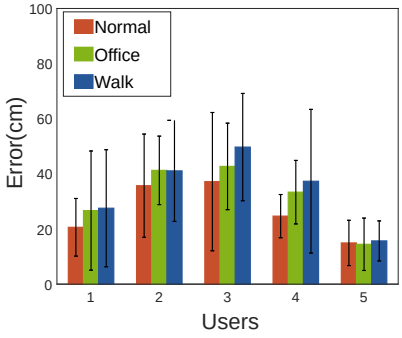


Figure 15. Coarse estimation error for different scenarios

that the coarse estimation algorithm is robust under different scenarios. The average errors of "Normal", "Office", and "Walk" are 0.27 m, 0.32 m, and 0.34 m, respectively. For most users the estimation accuracies are just slightly lower in the "Office" and "Walk" scenario. However, if the walking volunteer is very close to the user, the estimation performance would degrade severely.

The coarse estimation error is mainly due to two reasons: Firstly, the trajectories of CSI I/Q components caused by the hand movement are not perfect circles. When the trajectory deviates from circles, the phase differences of the dynamic vectors no longer equal to that of the tangent vectors as in Eq. 7 (*i.e.* $\Delta arg(\overrightarrow{H}_{tg}(t))_1^2 \neq \Delta arg(\overrightarrow{H}_{hd}(t))_1^2$). Secondly, there are still some remaining noises that make the phase measurements inaccurate. As shown in Figure 12, the large initial position estimation error may lead to deviated trace. Therefore, we need to further refine the initial position estimation.

4.3 Fine Initial Position Estimation

After estimating the initial position of hand at a coarse resolution, we select a small candidate region, *e.g.*, less than 0.5 m around the coarse position estimation as the candidate set for refinement. In this section, we focus on refining the resolution using a heuristic algorithm.

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 12, 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 refine the initial position in 2D space. For example, mTrack [29] 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 is dependent on the beamwidth of the directional antenna. With the directional antenna, the phase of a signal might not be significantly affected by the movement outside the transmitter's beam, which incurs degradations of tracking accuracy. Another ultrasonic work, LLAP, uses Inverse Discrete Fourier Transform to process CFR signals for all subcarriers to get 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.

Algorithm 2: Estimating Initial Position

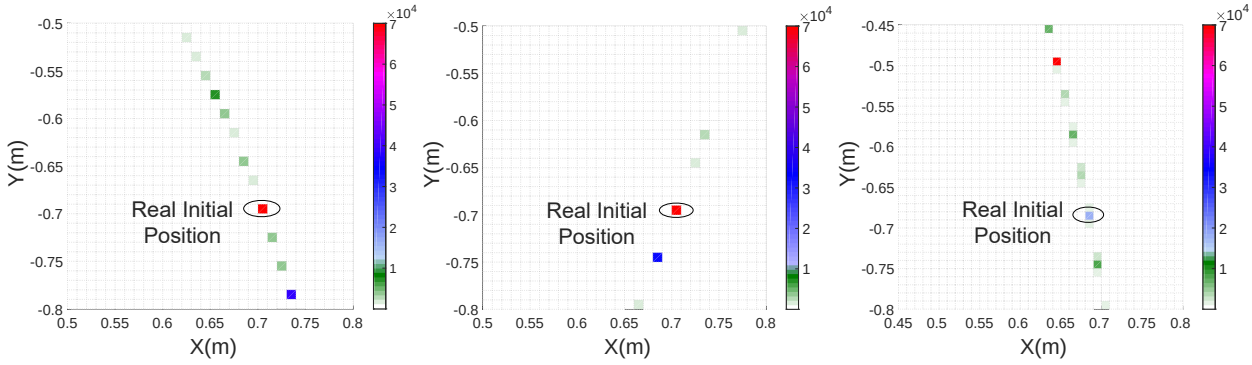
Input: Vertical movement path length changes of two receivers: $\Delta d_1^v, \Delta d_2^v$, horizontal movement path length changes of two receivers: $\Delta d_1^h, \Delta d_2^h$, coarse initial position (x_e, y_e)

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

for $k = 1$ *to* K **do**
- Determine the candidate square region: $C = \{(x_i, y_i) | ((k-1) * L < |x_i - x_e| \leq k * L), ((k-1) * L < |y_i - y_e| \leq k * L)\}$
- /*Vertical movement calculation*/
- Calculate the tracking trajectory $(\widehat{x_i^v}, \widehat{y_i^v})$ for two receivers based on the candidate initial position $(x_i^v, y_i^v) \in C$ and path change $\Delta d_1^v, \Delta d_2^v$;
- /*Horizontal movement calculation*/
- Calculate the tracking trajectory $(\widehat{x_i^h}, \widehat{y_i^h})$ for two receivers based on the initial position $(x_i^h, y_i^h) \in C$ 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 $(|\widehat{x_i^v} - x_i^v|)$ of X-axis;
- Find M candidate positions (x_i^h, y_i^h) which have the top M smallest deviations $(|\widehat{y_i^h} - y_i^h|)$ 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 $z_{M\times N}$;
- **if** $|\widehat{x_{j_{min}}^v} - x_{j_{min}}^v| < th_x$ *and* $|\widehat{y_{i_{min}}^h} - y_{i_{min}}^h| < th_y$ *and* $z_{i_{min}, j_{min}} < th_z$ **then**
  - $(x_0, y_0) \leftarrow (\frac{(x_{i_{min}}^h + x_{j_{min}}^v)}{2}, \frac{(y_{i_{min}}^h + y_{j_{min}}^v)}{2})$;
  - break;

return (x_0, y_0) ;

We refine 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 the hand along the 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.



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

Figure 16. Initial position simulation and experimental results

To better understand this method, we illustrate the process of initial position refinement via simulation. Suppose that a user moves his hand along X-axis for 30 cm. We use a maximum likelihood scheme to determine the initial position based on the measured distance change. Generally, the gesture trajectory (x, y) related to candidate initial position (x', y') can be determined by solving the following two equations:

$$\begin{cases} \sqrt{x'^2 + y'^2} + \sqrt{(x' - L_1)^2 + y'^2} \\ = \sqrt{x^2 + y^2} + \sqrt{(x - L_1)^2 + y^2} + \Delta d_1 \\ \\ \sqrt{x'^2 + y'^2} + \sqrt{x'^2 + (y' - L_2)^2} \\ = \sqrt{x^2 + y^2} + \sqrt{x^2 + (y - L_2)^2} + \Delta d_2 \end{cases} \tag{12}$$

where Δd_1 and Δd_2 are the path length changes corresponding to two receivers at $(L_1, 0)$ and $(0, L_2)$, while the source is at $(0, 0)$. Each path contains two segments, *e.g.*, $\sqrt{x'^2 + y'^2}$ (from source to candidate initial position) and $\sqrt{(x' - L_1)^2 + y'^2}$ (from the candidate initial position to RX1 at $(0, L_1)$). Eq.3 shows how the path length changes when the hand moves from candidate initial position (x', y') to (x, y). Therefore, the point (x, y) on the gesture trajectory is on the intersections of two elliptic curves determined by the position of receivers and the measured path length change. Figure 16(a) shows the simulating result for horizontal movement. The color represents deviation of the trajectory along Y-axis for various initial positions (*i.e.*, $\arg\max_{y'} \frac{1}{|y' - y|}$). Similar to the horizontal movement, Figure 16(b) shows deviation of the trajectory along X-axis for the vertical movement (*i.e.*, $\arg\max_{x'} \frac{1}{|x' - x|}$). From Figure 16(a) and Figure 16(b), we find that the red point is the real initial position in our simulation. However, in reality, the real initial position may not be the red point, since the distance measurement may have some errors. As shown in Figure 16(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 description of locating the initial point. We first select the candidate area around coarse position estimation (x_e, y_e) that is a square with side-length of L. Since the majority of the estimation error is less than 0.5 m as shown in Figure 13, we set L as 0.5 m. We use an iterative candidate region selection algorithm that converges until the final initial position is determined. We calculate the tracking trajectory corresponding to all positions in the candidate region to find a limited number of candidates for the two directions separately. These candidates should have movement deviations smaller than two empirical threshold th_x and th_y, respectively. We set both th_x and th_y as 0.05 m in the following experiments. Next, we find two closest candidate positions in the two directions and derive the new estimation by linearly combining these two positions. The iteration stops if the distance between these two candidate positions is smaller than an empirical threshold th_z, which is set as 0.1 m.

4.4 Successive 2D Tracking

As the lengths of the two initial paths from the transmitter to two receivers in Figure 12 (d_1 and d_2) are estimated in Section 4.3, the instantaneous length of the paths d_1' and d_2' can be calculated by updated distance change measurements (*i.e.*, Δd_1 and Δd_2). Then, by solving Eq.(11) with the conditions of $x > 0$ and $y < 0$, the realtime position of hand can be located, and the successive position will be derived from this iterative method.

4.5 Trajectory Correction

The estimated 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 KF based on a 2D Continous Wiener Process Acceleration (CWPA) model [31], 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

$$\boldsymbol{s}_k = [x_k \;\; y_k \;\; \dot{x}_k \;\; \dot{y}_k \;\; \ddot{x}_k \;\; \ddot{y}_k]^{\mathrm{T}} \tag{13}$$

where (x_k, y_k), $(\dot{x}_k, \dot{y}_k)$ and $(\ddot{x}_k, \ddot{y}_k)$ are the movement distances, velocity and acceleration of the hand at time k, respectively. The KF model assumes the true state at time k is evolved from the state at $(k-1)$ according to

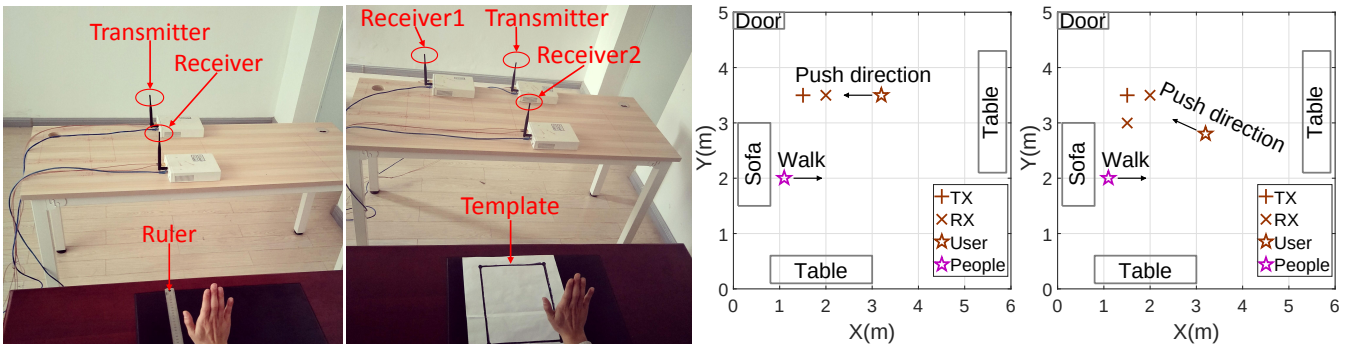
$$\boldsymbol{s}_k = \boldsymbol{A}\boldsymbol{s}_{k-1} + \boldsymbol{q}_k \tag{14}$$

where only the movement acceleration has 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

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & t & 0 & \frac{1}{2}t^2 & 0 \\ 0 & 1 & 0 & t & 0 & \frac{1}{2}t^2 \\ 0 & 0 & 1 & 0 & t & 0 \\ 0 & 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{15}$$

The measurement vector at time k in our system is

$$\boldsymbol{z}_k = [x_k \;\; y_k]^{\mathrm{T}} \tag{16}$$



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

Figure 17. Evaluation environment in laboratory

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

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{v}_k \tag{17}$$

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

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

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

Note that other trajectory correction methods (*e.g.*, Roughness Penalty Smoothing and particle filter) have the similar performance (as shown in Figure 20) to ours, they are not suitable for our system for their high computation cost and large processing delay.

5 Implementation and Evaluation

5.1 Implementation

We implemented WiTrace on the software radio platform-USRP-N210 hardware. The transmitting USRP with SBX board sends IEEE 802.11g OFDM frames [32] with bandwidth 20 MHz at 2.4 GHz [33]. While it is possible to use higher bandwidth to get better Time-of-Flight (ToF) measurements, we decide to use 20 MHz bandwidth because we mainly use the phase information for localization, which is more accurate than ToF and does not benefit much from a higher bandwidth. 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 a downsample rate of 100. Therefore, the sampling rate is reduced to $20\ \mathrm{M}/(64 * 100) = 3.125$ KHz for each subcarrier. The transmission power is set to 20 dBm which is the same as the COTS WiFi NIC. Note that the major difference between the CSI from commercial WiFi devices and USRP is that the USRP uses an external clock to synchronize the transceiver system, which can effectively reduce the CFO, while CFO can only be reduced by heuristic clock compensation algorithms in WiFi CSI [13]. Thus, the signal phase measured by USRP is more stable than by commercial WiFi devices. For 1D tracking, our system uses one receiving USRP, as shown in Figure 17(a). For 2D tracking, two receiving USRPs are placed on the table as shown in Figure 17(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, are at the height of 0.8 m, and the user's hand is at the same height. 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 evaluation results are processed offline using MATLAB. We choose offline evaluation because it is more repeatable in comparing the tracking performance between various schemes than real-time experiments.

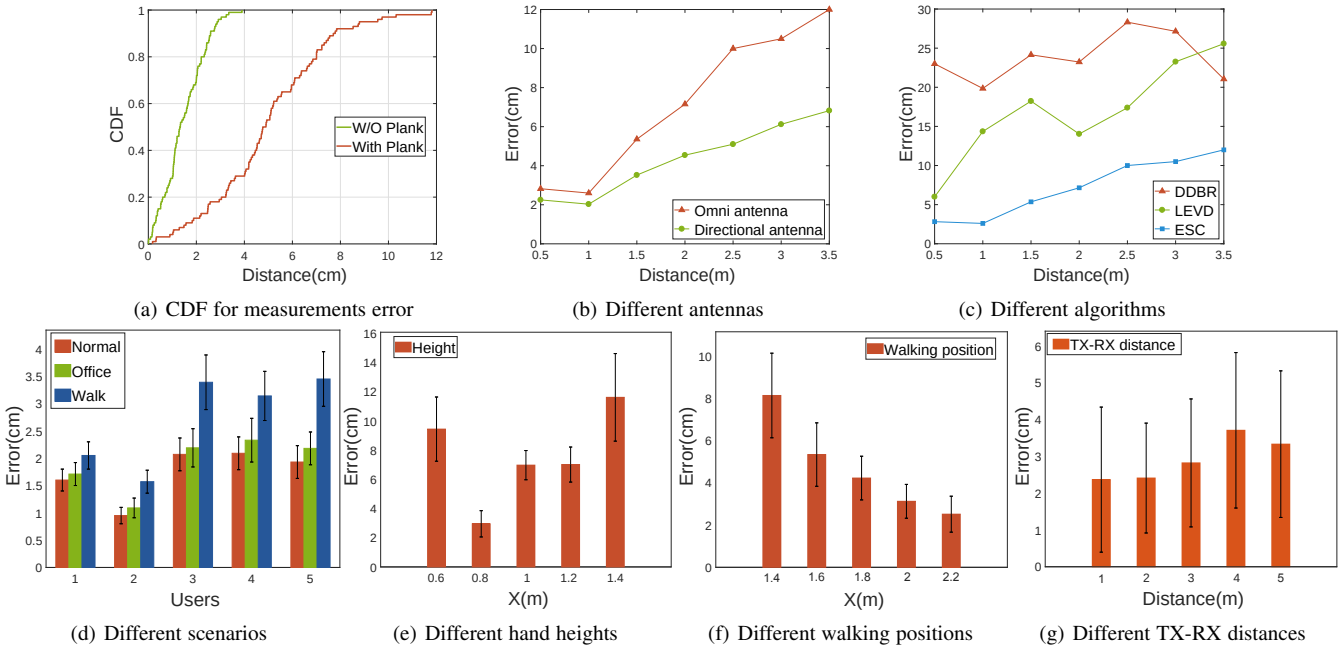
5.2 Evaluation Metrics

Our experiments are conducted in a laboratory with an area of 5 m $\times$ 6 m. For 1D scenario, the transmitter and the receiver are set in a line, as shown in Figure 17(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 17(d). We evaluated 2D tracking with omnidirectional antennas from four 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*.

5.3 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 17(c), the initial position of volunteer is 1.2 m away from the receiver and the volunteer pushes hand



(a) CDF for measurements error (b) Different antennas (c) Different algorithms

(d) Different scenarios (e) Different hand heights (f) Different walking positions (g) Different TX-RX distances

Figure 18. 1D tracking error

for a distance of 30 cm towards the receiver. Figure 18(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.

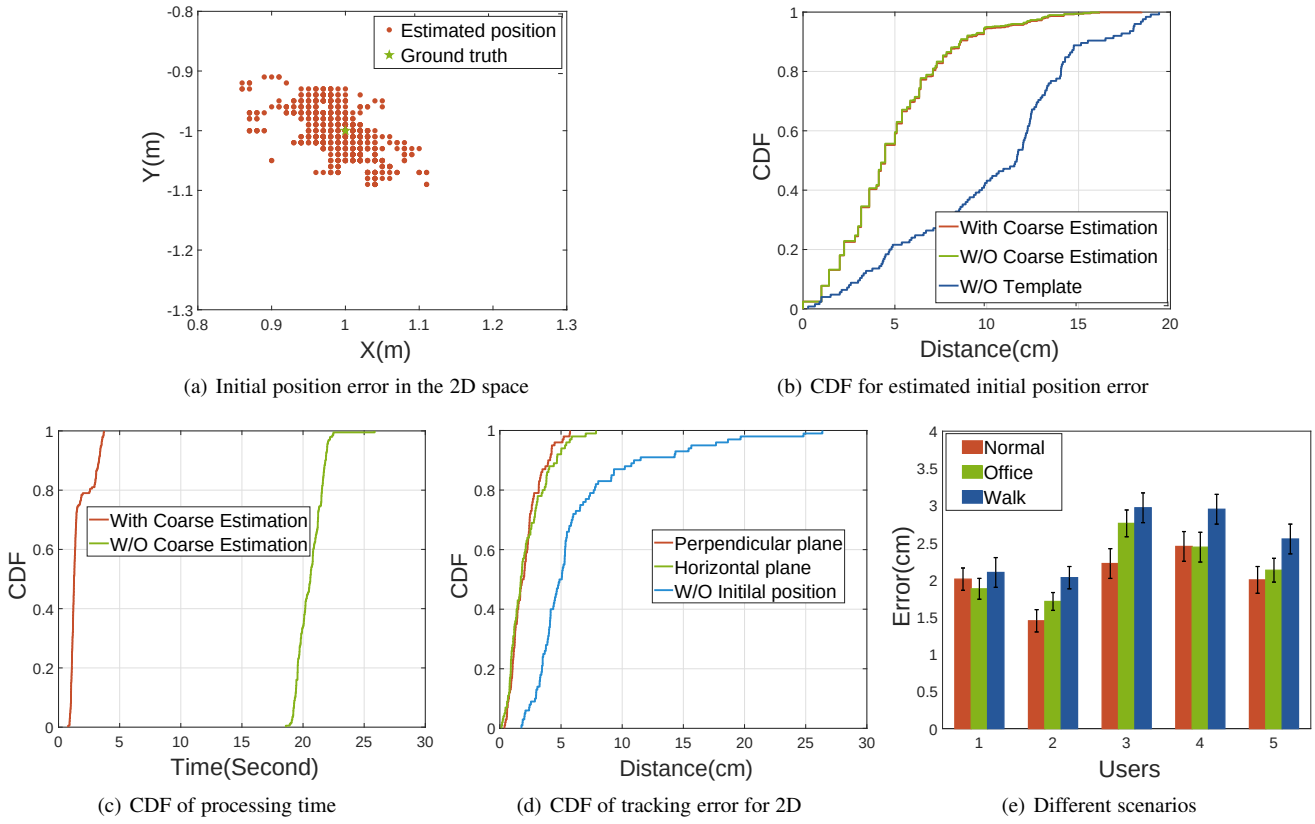
Tracking accuracy with different antennas in 1D space: *WiTrace achieves average error of* 4.48 *cm and* 3.09 *cm when using omnidirectional antennas and directional antennas in the range of* 2 *m, respectively.* Figure 18(b) shows the average error by using omnidirectional antennas and directional antennas when a volunteer is at different distances from the receiver. The results show that WiTrace achieves an average distance error of 4.48 cm and 3.09 cm when the hand moves for 30 cm over a distance of less than 2 m for omnidirectional antenna and directional antenna, respectively. The average tracking errors for both omnidirectional antenna and directional antenna will increase to 7.20 cm and 4.34 cm within the range of 3.5 m, due to the reduced SNR.

Note that the 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 omnidirectional antenna obviously. 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 18(c), the average errors are 7.20 cm, 16.97 cm and 23.82 cm by using ESC, LEVD, and DDBR in the range of 3.5 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.

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 with age from 20 to 62. As shown in Figure 18(d), the standard deviation of different users is 0.86 cm, which shows WiTrace is robust for different users. Figure 18(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 walking 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. Our system's robustness against walking interferences is due to two reasons. Firstly, most high frequency noises caused by the walking movement can be efficiently removed through the low-pass filter in Section 3.1. Secondly, since the signal strength decays with the distance according to the power law, the reflection of walking volunteer at 2 m has a limited effect on the user's pushing movement within 1.2 m. However, if the walking people is very close to the pushing user, the tracking performance would degrade severely.

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 18(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, closer distance between another walking volunteer and the receiver will decrease the SNR of our system shown in Figure 18(f). The lower SNR will make it more difficult to detect movement of hand accurately, leading to larger tracking errors. The result above



(a) Initial position error in the 2D space

(b) CDF for estimated initial position error

(c) CDF of processing time

(d) CDF of tracking error for 2D

(e) Different scenarios

Figure 19. 2D measurement

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.

Impact of TX-RX distance in 1D space: *WiTrace achieves an average tracking error that is lower than 3.71 cm over different TX-RX distances.* We change the TX-RX distances from 1 m to 5 m and ask the user to push for 30 cm at a distance of 1.2 m to the receiver. Figure 18(g) shows that WiTrace achieves the best tracking performance (*i.e.*, 2.37 cm error on average) with a TX-RX distance of 1 m. With the TX-RX distance of larger than 3 m, WiTrace undergoes a slight degradation trend.

Estimated initial position error in 2D space: *WiTrace achieves average* 6.23 *cm estimated error with the template, and average* 10.18 *cm error without template in 2D space.* Figure 19(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 19(b) shows that 80th percentile estimated distance error is within 9.48 cm with template. This is mainly because the dynamic path measurements have slight 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 previous result. This is because the movement of user's hand is more random than pushing along the template. Figure 19(b) also shows that the initial estimation errors with and without the coarse estimation are almost the same.

Computational complexity of initial position estimation: *WiTrace's initial position estimation process reduces the computational complexity by more than 12 times when compared to our preliminary algorithm [1].* We evaluate the computational complexity of our algorithm by running the algorithm on the same dataset with 150 instances on a desktop with an i7-7700 CPU. The average running time of the two-step initial position estimation is 1.65 seconds, while our preliminary algorithm without coarse estimation takes 20.58 seconds on the same machine. The superior performance is mainly due to the smaller candidate region provided by the coarse estimation.

Tracking error in 2D space: *For 2D tracking, WiTrace achieves an average tracking error of* 2.09 *cm.* Figure 20 shows samples of three shapes' trajectories (*i.e.*, rectangle, triangle, and circle) drawn by WiTrace. We calculate each trajectory with time interval of 0.04 seconds for adjacent points. The average time for users to finish drawing the rectangle, triangle, and circle are 2.9 seconds, 2.3 seconds, and 2.1 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 19(d) 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 17(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 19(d), 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 20 also shows that Kalman Filter improves the tracking accuracy effectively.

Additionally, we evaluate WiTrace's performance when the user draws more complicated trajectories. As shown in Figure 21 , WiTrace can track handwritten letters (*i.e.*, "t", "m", "c") with high accuracy.

Robustness for different scenarios and users: *WiTrace achieves tracking error of less than* 2.98 *cm for different people and scenarios.* Figure 19(e) shows the mean error of estimated trace for different users across 50 drawings for each user under 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 both small and large movements.

Impact of pushing direction: *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 12, users are invited to push hands in straight lines for 30 cm with different degrees related to the RX2. Figure 22 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.

6 Related Work

6.1 RF Based Gesture Recognition and Tracking

Recently, WiFi signal measurements such as CSI, RSSI are used for gesture recognition [3]–[5], [13]. WiKey proposes to use CSI dynamics to recognize keystrokes [3]. WiFinger uses CSI to recognize a set of eight gestures with accuracy of 93% [4]. 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% [9]. 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 [10], [14], [15], [29], [34], [35]. Unfortunately, these systems still have some drawbacks. some schemes require users to wear RF devices [34], [36], 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 [34]. Some schemes have a limited tracking range, such as WiDraw [10] 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 [35]. However, they need to use antenna arrays and their mean localization error is 0.8 m. Our recent work QGesture [13] achieves a hand tracking accuracy of 5.5 cm using the coarse phase information extracted from commercial WiFi devices. However, it focuses on quantifying the gesture distance and coarse direction based on known initial positions. In contrast, WiTrace develops a set of CSI-based schemes to estimate the initial position with an average accuracy of 6.23 cm. Secondly, WiTrace realizes successive 2D gesture tracking with fine-grained accuracy followed by trajectory correction based on Kalman Filter.

6.2 Non-RF Based Gesture Recognition and Tracking

Non-RF gesture recognition mainly includes vision based [6], [7], [37]–[40] and sound based [5], [16], [19], [41], [42]. Vision based systems incur high computational cost and highly depends on the viewing angle and lighting conditions [6], [7], [38]–[40]. Sound-based system, LLAP [16], 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 [19] proposes 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.

7 Limitations and Future Work

WiTrace establishes the feasibility of using OFDM signals to track gesture. However, our current implementation has some limitations. First, we currently calibrate the trajectory using our initial position estimating algorithm which requires the 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 further 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 [43], we plan to implement the system on the commercial WiFi NICs instead of software radio devices in our future work. Third, it is challenging to implement 3D tracking based on the current approach of WiTrace. This is because when the hand is moving in the 3D space, the reflection of other body parts is so strong that isolating the reflection from the hand becomes difficult for WiTrace.

8 Conclusions

In this paper, we made three key contributions. First, we utilize the phase changes of OFDM signal to achieve high accuracy gesture tracking. Second, to enable the 2D tracking, we propose a new 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 conduct comprehensive evaluation. Our experimental results show that WiTrace achieves cm-level tracking accuracy at a distance of 3.5 meters.

9 Acknowledgement

This work was supported in part by the National Key R & D Program of China under Grant No. 2018YFB1004704. In part by the National Natural Science Foundation of China under Grant 61872178, 61502229, 61832005, 61672276, 61872173, and 61321491. In part by the Fundamental Research Funds for the Central Universities under Grant 021014380079.

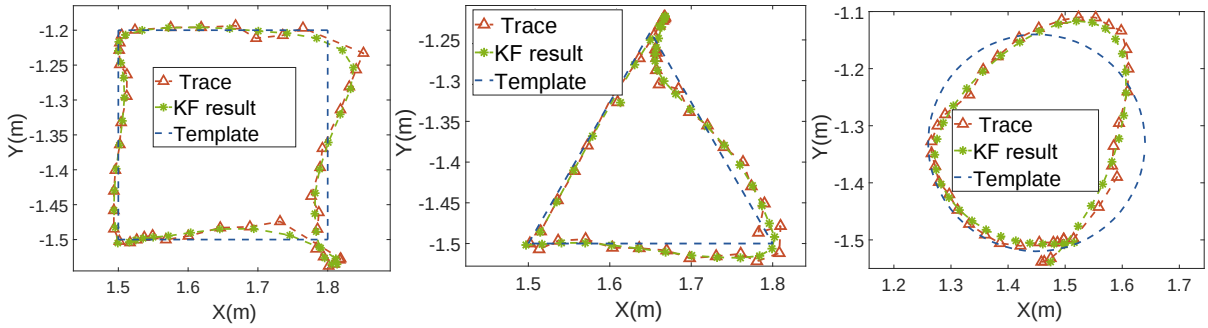
References

[1] L. Wang, K. Sun, H. Dai, A. X. Liu, and X. Wang, "Witrace: Centimeter-level passive gesture tracking using wifi signals," in *Proc. IEEE SECON*, 2018.

[2] K. Sun, W. Wang, A. X. Liu, and H. Dai, "Depth aware finger tapping on virtual displays," in *Proc. ACM MobiSys*, 2018.

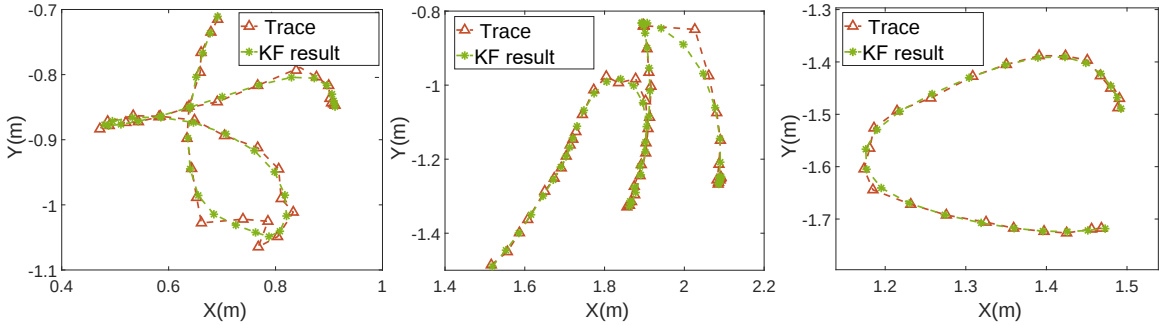
[3] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," in *Proc. ACM MobiCom*, 2015.

[4] S. Tan and J. Yang, "WiFinger: leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. ACM Mobihoc*, 2016.



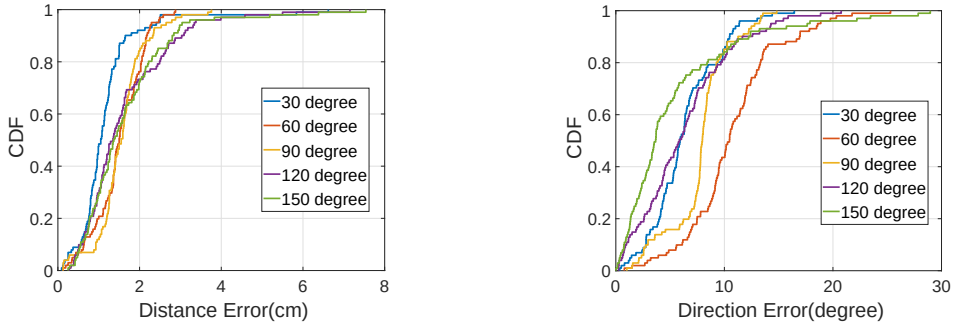
(a) Rectangle (b) Triangle (c) Circle

Figure 20. 2D trace for different shapes



(a) Letter "t" (b) Letter "m" (c) Letter "c"

Figure 21. Trace of three letters



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

Figure 22. 2D trace errors with various directions

[5] 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.

[6] "Microsoft Kinect," http://www.microsoft.com/en-us/kinectforwindows/.

[7] "Leap Motion," https://www.leapmotion.com/.

[8] S. Nirjon, J. Gummeson, G. Dan, and K. H. Kim, "Typingring: A wearable ring platform for text input," in *Proc. ACM Mobisys*, 2015.

[9] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi based gesture recognition system," in *Proc. IEEE INFOCOM*, 2015.

[10] 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.

[11] 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.

[12] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive human tracking with a single Wi-Fi link," in *Proc. ACM MobiSys*, 2018.

[13] N. Yu, W. Wang, A. X. Liu, and L. Kong, "Qgesture: Quantifying gesture distance and direction with WiFi signals," *Proc. ACM UbiComp*, 2018.

[14] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. Usenix NSDI*, 2013.

[15] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person motion tracking via RF body reflections," in *Proc. Usenix NSDI*, 2015.

[16] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. ACM MobiCom*, 2016.

[17] Y. W. Huijie Chen, Fan Li, "Ecotrack: Acoustic device-free hand tracking on smart phones," in *Proc. IEEE INFOCOM*, 2017.

[18] H. Z. L. Q. Sangki Yun, Yi-chao Chen and W. Mao, "Strata: Fined-grained device-free tracking using acoustic signals," in *Proc. ACM MobiSys*, 2017.

[19] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proc. ACM CHI*, 2016.

[20] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, "Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient," in *Proc. ACM UbiComp*, 2017.

[21] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," in *Proc. IEEE INFOCOM*, 2012.

[22] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, 2012.

[23] "Enhancements for higher throughput," IEEE Standard 802.11n, 2009.

[24] J. G. Proakis, "Digital communications," *McGraw-Hill*, 1995.

[25] "OctoClock-G," https://www.ettus.com/product/details/OctoClock-G.

[26] 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.

[27] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. ACM MobiCom*, 2015.
[28] L. Davies and U. Gather, "The identification of multiple outliers," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 782–792, 1993.
[29] T. Wei and X. Zhang, "mTrack: High-precision passive tracking using millimeter wave radios," in *Proc. ACM MobiCom*, 2015.
[30] 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.
[31] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
[32] "IEEE 802.11g-2003," https://en.wikipedia.org/wiki/IEEE_802.11g-2003.
[33] IEEE computer society LAN MAN standards committee, "IEEE Std 802.11-1999," *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999.
[34] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: virtual touch screen in the air using RF signals," in *Proc. ACM SIGCOMM*, 2014.
[35] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using RF backscatter," in *Proc. Usenix NSDI*, 2015.
[36] 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.
[37] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *Proc. ACM MobiCom*, 2015.
[38] 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.
[39] "Microsoft Hololens," http://www.microsoft.com/microsoft-hololens/en-us.
[40] X. A. Chen, J. Schwarz, C. Harrison, J. Mankoff, and S. E. Hudson, "Air+touch: interweaving touch & in-air gestures," in *Proc. ACM UIST*, 2014.
[41] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proc. ACM CHI*, 2012.
[42] K. Ling, H. Dai, Y. Liu, and A. X. Liu, "Ultragesture: Fine-grained gesture sensing and recognition," in *Proc. IEEE SECON*, 2018.
[43] Y. Zhuo, H. Zhu, H. Xue, and S. Chang, "Perceiving accurate CSI phases with commodity WiFi devices," in *Proc. IEEE INFOCOM*, 2017.

Lei Wang is a fourth year PhD student in Department of Computer Science and Technology at Nanjing University, China. His research interests are in the areas of wireless networking, including Device-free Sensing, Software Defined Radio and Smartphone Sensors.

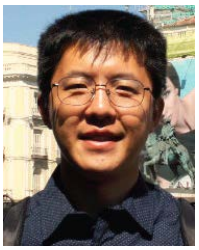
Ke Sun Ke Sun received his B.S. degree in Computer Science from Nanjing University of Aeronautics and Astronautics, Jiangsu, China, in 2016. He is currently a Master student in Nanjing University. His research interests are in the area of mobile computing.

Haipeng Dai received the B.S. degree in the Department of Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the Ph.D. degree in the Department of Computer Science and Technology in Nanjing University, Nanjing, China, in 2014. His research interests are mainly in the areas of wireless charging, mobile computing, and data mining. He is a research assistant professor in the Department of Computer Science and Technology in Nanjing University. His research papers have been published in many prestigious conferences and journals such as ACM MobiSys, ACM MobiHoc, ACM VLDB, ACM SIGMETRICS, ACM Ubicomp, IEEE INFOCOM, IEEE ICDCS, IEEE ICNP, IEEE SECON, IEEE IPSN, IEEE JSAC, IEEE/ACM TON, IEEE TMC, IEEE TPDS, and IEEE TOSN. He is an IEEE and ACM member. He serves/ed as Poster Chair of the IEEE ICNP'14, Track Chair of the ICCCN'19, TPC member of the IEEE ICNP'14, IEEE ICC'14-18, IEEE ICCCN'15-18 and the IEEE Globecom'14-18. He received Best Paper Award from IEEE ICNP'15, Best Paper Award Runner-up from IEEE SECON'18, and Best Paper Award Candidate from IEEE INFOCOM'17.

Wei Wang received his MS and Ph.D. degree from the ESE department of Nanjing University and the ECE department of National University of Singapore, in 2000 and 2008 respectively. He is currently an Associate Professor in the CS department of Nanjing University. His research interests are in the area of wireless networks, including Device-free Sensing, Cellular Network Measurements, and Software Defined Radio systems.

Kang Huang received the Bachelor degree of Bell Honor School from the Nanjing University of Posts and Telecommunications, China in 2016. Master candidate of Computer Technology oriented in information security from Nanjing University.

Xiaoyu Wang received the B.S. degree in the Department of Computer Science and Technology from Soochow University, Suzhou, Jiangsu, China, in 2016. She is studying towards the Ph.D degree in the Department of Computer Science and Technology in Nanjing University. Her research interests focus on wireless charging and data mining.



Alex X. Liu received his Ph.D. degree in Computer Science from The University of Texas at Austin in 2006. He received the IEEE & IFIP William C. Carter Award in 2004, a National Science Foundation CAREER award in 2009, and the Michigan State University Withrow Distinguished Scholar Award in 2011. He is an Associate Editor of IEEE/ACM Transactions on Networking, an Editor of IEEE Transactions on Dependable and Secure Computing, and an Area Editor of Computer Communications. He received Best Paper Awards from ICNP-2012, SRDS-2012, and LISA-2010. His research interests focus on networking and security.

Qing Gu received his Ph.D. degree in computer science from Nanjing University, Nanjing. He is a professor of the State Key Laboratory of Novel Software Technology, and the Department of Computer Science and Technology, Nanjing University, Nanjing. His research interests include software testing, quality and process improvement, software maintenance and evolution, and complex network.