WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices

Li Sun
University at Buffalo, SUNY
lsun3@buffalo.edu

Souvik Sen
HP Labs
souvik.sen@hp.com

Dimitrios Koutsonikolas
University at Buffalo, SUNY
dimitrio@buffalo.edu

Kyu-Han Kim
HP Labs
kyu-han.kim@hp.com

ABSTRACT

This paper demonstrates that it is possible to leverage WiFi signals from commodity mobile devices to enable hands-free drawing in the air. While prior solutions require the user to hold a wireless transmitter, or require custom wireless hardware, or can only determine a pre-defined set of hand gestures, this paper introduces *WiDraw*, the first hand motion tracking system using commodity WiFi cards, and without any user wearables. *WiDraw* harnesses the Angle-of-Arrival values of incoming wireless signals at the mobile device to track the user's hand trajectory. We utilize the intuition that whenever the user's hand occludes a signal coming from a certain direction, the signal strength of the angle representing the same direction will experience a drop. Our software prototype using commodity wireless cards can track the user's hand with a median error lower than 5 cm. We use *WiDraw* to implement an in-air handwriting application that allows the user to draw letters, words, and sentences, and achieves a mean word recognition accuracy of 91%.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; H.5.2 [**User Interfaces**]: Input devices and strategies

General Terms

Design, Experimentation, Performance

Keywords

Wireless; Motion Tracking; Gesture Recognition; Channel State Information; Angle-of-Arrival

1. INTRODUCTION

Today, a user can write, scroll, swipe, or draw on a touch screen of a smart phone, tablet, or laptop. Users can even interact hands-free with a computing device using commercial hardware systems such as Kinect [1] and Leap Motion [2]. While these systems are highly popular, they require a dedicated hardware setup or line-of-sight (LOS) to the user. Recently, researchers have made progress in detecting human hand gestures using wireless signals [3–8], enabling gesture recognition in non-line-of-sight (NLOS) scenarios. However, prior solutions are limited because they either require the user to hold a wireless transmitter [7], or require a custom wireless hardware to track the user's hand motions [3–6, 8]. More importantly, most of the current solutions [3–5, 8] require a priori learning of wireless signal patterns, which limits them to recognize only a fixed set of pre-defined hand gestures. We believe that, if we can leverage wireless signals to automatically track the detailed trajectory of the human hand in the air, we can greatly simplify interaction of the user with today's computing devices including laptops, smart TVs, tablets, and even devices that do not have a dedicated user input (e.g., sensors and IoT devices). Since WiFi signals do not require LOS and can traverse through material (e.g., cloth), hands-free drawing can also be used by the user to interact with the mobile device in a pocket or a bag - e.g., to control volume or answer a call, or simply draw the name of the callee in the air to start a conversation.

This paper demonstrates that it is possible to leverage WiFi signals from commodity mobile devices to enable hands-free drawing in the air. We introduce *WiDraw*, the first hand motion tracking solution that can be enabled on existing mobile devices using only a software patch. *WiDraw* leverages physical layer information and multiple antennas on commodity devices to track the detailed trajectory of the user's hand in both LOS and NLOS scenarios, without requiring the user to touch the device or hold any hardware. As shown in Figure 1(a), using *WiDraw*, a user can draw arbitrary lines, curves, or even alphabetical characters, simply by using hand motions in the air.

WiDraw utilizes the Angle-of-Arrival (AoA) values of incoming wireless signals at the mobile device to track the user's hand trajectory. Existing techniques such as MUSIC [9] can utilize the *Channel State Information* (CSI) from multiple antennas to estimate the angles at which a wireless signal from a transmitter arrives at the receiver. MUSIC also reports the signal strength of the incoming signal component arriving from a given angle. We utilize the intuition that whenever the user's hand blocks a signal coming from a certain direction, the signal strength of the AoA representing the same direction will

MobiCom'15, September 7–11, 2015, Paris, France.

DOI: http://dx.doi.org/10.1145/2789168.2790129.

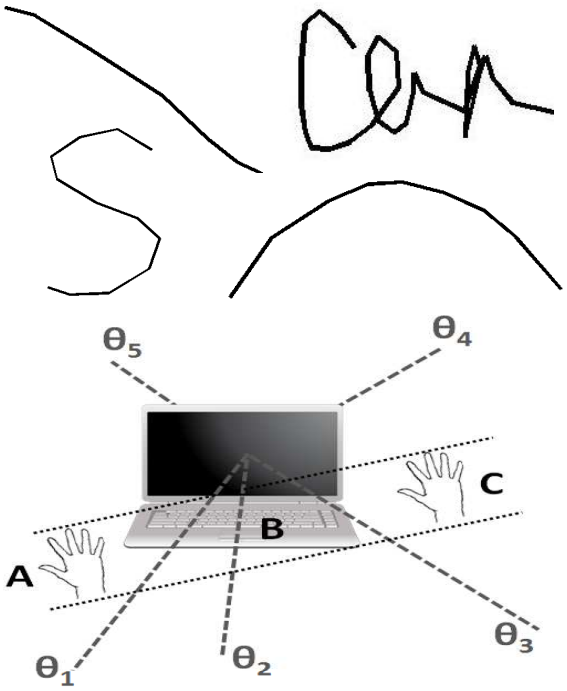

Figure 1: (a) *WiDraw*'s estimated trajectory when the user drew a straight line, a circular curve, the letter "S", and the word "can". (b) The user's hand perturbs the signal strength of several AoAs along its trajectory.

experience a sharp drop. E.g., in Figure 1(b), if the hand passes through the trajectory ABC, it will occlude the AoAs θ_1, θ_2, and θ_3 in sequence. By tracking the signal strength of the AoAs, it is possible to determine when such occlusions happen. Thereafter, by utilizing the azimuth and elevation of the affected AoAs, it is possible to determine a set of horizontal and vertical coordinates along the hand's trajectory. The depth of the user's hand can also be approximated using the drop in the overall signal strength. E.g., if the hand is close to the receiver, it will occlude a larger number of AoAs in comparison to when the hand is farther away from it. By estimating the hand's depth, along with horizontal and vertical coordinates, *WiDraw* tracks the hand's trajectory w.r.t. the WiFi antennas of the receiver.

We demonstrate the feasibility of *WiDraw* by building a software prototype on HP Envy laptops, using Atheros AR9590 chipsets and 3 antennas. The tracking granularity of *WiDraw* depends on the number of AoAs along the hand's trajectory. While periodic access point beacons can contribute several angles, the tracking granularity can be further increased by employing lightweight probing mechanisms to obtain AoAs from neighboring client devices as well. We show that by utilizing the AoAs from up to 25 WiFi transmitters, the *WiDraw*-enabled laptop can track the user's hand with a median error lower than 5 cm. *WiDraw*'s rate of false positives - motion detection in the absence of one - is less than 0.04 events per minute over a 360 minute period in a busy office environment. Experiments across 10 different users also demonstrate that *WiDraw* can be used to write words and sentences in the air, achieving a mean word recognition accuracy of 91%.

Our main contributions are summarized as follows:

- ***WiDraw* is the first hand motion tracking system using commodity WiFi cards:** Existing systems can only track a few gestures, require special hardware, or require the user to hold a device.
- **We identify the opportunity to utilize the AoA information to enable hands-free drawing in the air:** Our solution harnesses the effect of hand occlusions on the signal strength of incoming signal directions.
- **We implement and demonstrate our solution using commodity wireless cards:** We exploit the CSI information from the three antennas of the Atheros AR9590 chipset and evaluate our system using more than 90,000 trajectories drawn by 10 users.
- **We design and implement a virtual handwriting application using WiDraw:** Our system achieves an average letter and word recognition accuracy of 95% and 91%, respectively.

The subsequent sections expand on each of these contributions, including experimental observations, followed by algorithms, design, and evaluation.

2. BACKGROUND AND OBSERVATIONS

This section presents the relevant background on AoA estimation followed by our observations that suggest that AoAs can be used to estimate the motion of the human hand in a 3D space. For our measurements, we use HP laptops running Linux OS. We note that AoA estimation is possible from commodity chipsets if they can report per packet *Channel State Information (CSI)* [10,11]. Next we briefly describe the mechanism to obtain AoAs from the CSI.

2.1 AoA estimation from CSI

A wireless signal from a transmitter arrives at several angles at the receiver. The AoA at the receiver can be computed by comparing the phase of the CSI values acquired from multiple antennas, a quantity which changes linearly by 2π for every wavelength λ along the path from the transmitter to the receiver. E.g., if the receiver has only two antennas placed at a distance of $\lambda/2$ and a measured phase difference of the CSI acquired from the two antennas is $\Delta\phi$, we can estimate the angle-of-arrival θ as:

$$\theta = \arcsin(\Delta\phi/\pi)$$

Rather than analyzing the phase differences between every pair of antennas individually, the receiver can employ AoA estimation algorithms such as MUSIC [9]. MUSIC utilizes the phase difference values and yields a pseudospectrum (Figure 2(a)). Each peak in the pseudospectrum is an estimated *AoA*. Multiple peaks in Figure 2(a) imply incoming signals along different directions from the transmitter.

Since wireless signals propagate in a 3-D space, the angle of an incoming signal at the receiver has both azimuth and elevation components (Figure 2(b)). MUSIC can compute both the azimuth and elevation of an AoA if the receiver's antennas are arranged in a special pattern [12]. However, when applied to linear antenna arrays such as those within a laptop or tablet screen, MUSIC can only compute a one-dimensional representation of the AoA (θ in Figure 2(b)). The 1-D AoA value θ, as reported by MUSIC, is related to the azimuth (az) and elevation (el) values of the AoA:

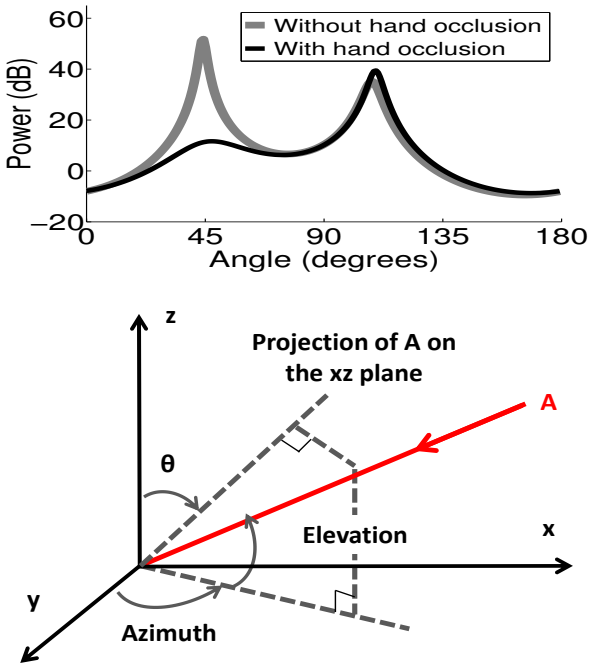

Figure 2: (a) MUSIC's pseudospectrum output with and without occlusion by human hand. (b) MUSIC yields the 1-D angle-of-arrival (AoA) which is related to the azimuth and elevation of the AoA.

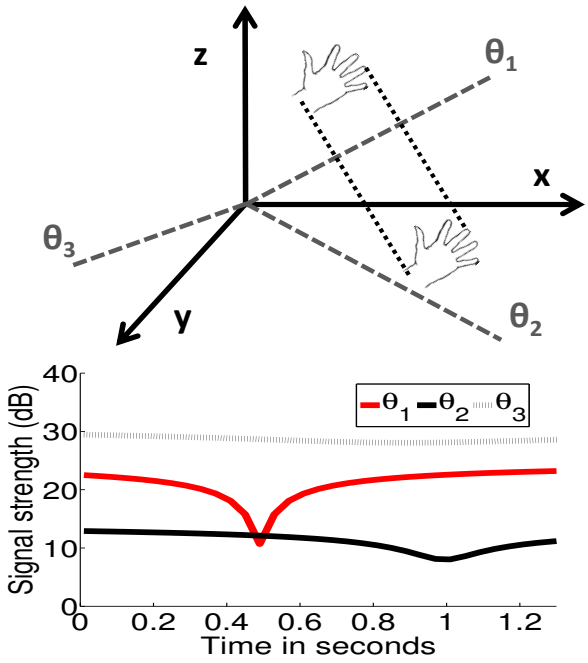

Figure 3: (a) Trajectory of human hand through 3 AoA values. (b) Change in amplitude of the AoAs as the human hand passes through them.

$$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{\tan(el)}{\sin(az)}\right) \quad (1)$$

Observe that θ is upper bounded by $(\frac{\pi}{2} - el)$, which occurs when the azimuth angle is $\frac{\pi}{2}$. In section 3.1, we will use these observations to compute both the azimuth and elevation values of the AoA from the existing linear antenna array of the laptop.

2.2 Effect of human hand on the AoA

To track the motion of a human hand near the wireless receiver, we observe that if the user's hand blocks the signal arriving along a specific AoA, the AoA's signal strength will experience a sharp drop (as shown in Figure 2(a)). On the other hand, if the trajectory of the user's hand does not occlude the signal arriving on a specific AoA, its signal strength will not demonstrate any drop. To illustrate the above, consider 3 different AoA values as observed by the receiver in Figure 3(a). The trajectory of the user's hand blocks θ_1 completely and θ_2 partially but does not pass through θ_3. Figure 3(b) shows the change in signal strength values of the 3 angles over time. The time instant when the signal strength of a given AoA attains its minimum value corresponds to the moment when the user's hand arrives at a location that is closest to that angle on its moving trajectory. We find that the drop in the signal strength of a given AoA value is proportional to proximity of the hand to the AoA trajectory, which is why θ_1 demonstrates a larger drop than θ_2 in Figure 3(b).

The set of AoAs whose signal strength values attain a minimum provides a mechanism to determine the hand's position and motion. If the user's hand is y_i units away from receiver and it causes an AoA with azimuth (az) and elevation (el) to attain a minimum, the horizontal (x_i) and vertical (z_i) coordinates of the user's hand can be computed as:

$$x_i = \frac{y_i}{\cot(az)} \qquad z_i = \frac{y_i * \tan(el)}{\cos(az)} \quad (2)$$

The trajectory of the human hand can be estimated by using the sequence of coordinates, as obtained from the AoAs whose signal strength drops. Of course the tracking accuracy depends on the number of angles along the hand's trajectory. A large number of angles placed close to each other along the hand's trajectory can allow tracing the trajectory with fine granularity. On the other hand, a small number of angles far apart from each other, along the hand's trajectory, will result in low accuracy. Therefore, the feasibility and limitations of hand tracking depend on the distribution of AoA values in real environments. Next, we study these requirements in detail.

2.3 AoA density for hand motion tracking

As shown in Figure 4(a), due to their radial nature, the spatial distance between incoming signal directions increases with larger distance from the WiFi antennas. Near the receiver, the density of the AoAs is quite high implying that the user's hand can be tracked with high granularity. However, farther the user's hand is from the receiver, lower will be the granularity of trajectory tracking. The density of the AoAs in a given environment in turn depends on the number of transmitters within the range of the *WiDraw*-enabled receiver. Figure 4(b) plots the average number of access points (APs) and WiFi-enabled devices across 50 different locations in different environments - apartment, office, cafeteria. To track the movement of the user's hand, *WiDraw* needs to periodically obtain

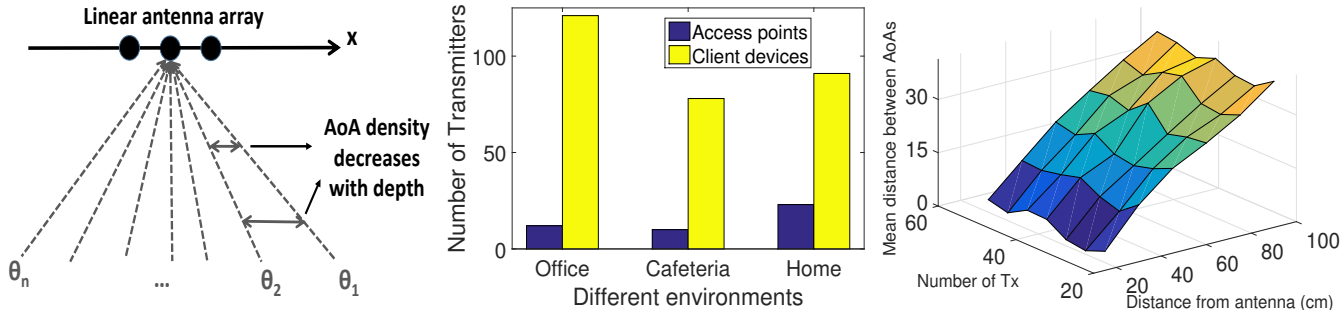

Figure 4: (a) Distance between nearby angles increases with greater distance from the receiver. (b) Average number of APs and WiFi-enabled devices across 50 different locations in different environments. (c) The average distance between neighboring AoAs increases with lower number of WiFi transmitters and increasing depth of the hand from the receiver.

and analyze the incoming signal directions of each transmitter. If the transmitter is an AP, *WiDraw* can obtain the CSI and AoA of its transmitted signal by leveraging periodic beacons. However, if the transmitter is a client device, the *WiDraw*-enabled receiver may have to periodically probe the device to obtain its AoA values. Therefore, *WiDraw* may only be able to leverage a few neighboring clients.

To estimate the tracking granularity using angles, we estimated the AoA values from neighboring transmitters whose signals are incident on the receiver at the frontside, using the data from the same 50 locations as in Figure 4(b). We consider the AoAs only at the front side of the receiver because the user will likely draw in front of the receiver. Figure 4(c) plots the average distance between nearby AoAs as a function of the number of transmitters and increasing distance from the laptop. We observe that for a modest number of transmitters (e.g., 30), the inter-AoA distance values are reasonably low (12.75 cm) for a distance of 1 foot but increases to greater than 24.65 cm for distances more than 2 feet. This suggests that at a larger distance from the receiving antennas, we will only be able to track coarse hand motion trajectories. However, as long as the human hand is within 2 feet from the receiver, the density of the AoAs may be reasonable enough to allow us to track finer hand motions.

2.4 Effect of environmental changes on AoA

AoA values need to be reasonably stable at least during the motion of the user's hand. The AoA values can change rapidly if the transmitter is mobile, making it difficult to use it for motion detection [11]. On the other hand, if the transmitter is static, an AoA will change only if a dominant propagation path between the transmitter and the receiver is affected. We find that the changes in signal strength of the AoAs due to environmental changes are structurally different from those because of the user's hand near the receiver. First, as shown in Figure 5(a), unlike hand occlusions, signal strength variations due to environmental fluctuations are often random and therefore can be removed using a low pass filter. Second, after filtering, the drop in signal strength due to environmental changes is significantly lower than when a human hand occludes the AoA (Figure 5(b)). Third, while hand occlusions result into a large drop in signal strength within a few seconds, such large changes occur much more slowly due to environmental fading. Therefore, it may be possible to increase robustness by estimating the hand's trajectory from the AoAs

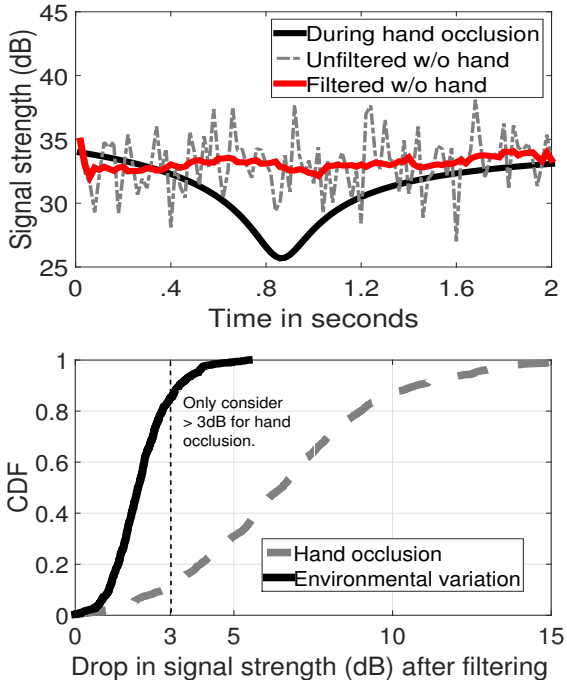

Figure 5: Signal strength fluctuations in a busy cafeteria: (a) Drop in signal strength over time is significantly lower than that due to occlusion from human hand. (b) Distribution of signal strength fluctuations at 50 different locations with and without hand occlusions.

whose signal strength demonstrates a large drop. Of course, our system requires more robust mechanisms if any particular AoA's signal strength changes significantly.

In conclusion, we observed that human hand causes significant drop in the signal strength of the AoAs along its trajectory. We found that the occlusion due to human hand causes deeper perturbations than random environment related fluctuation. Therefore, by tracking the azimuth and elevation of the AoAs which undergo significant changes in signal strength, it may be possible to determine the trajectory of the human hand. However, as observed in our measurements, several challenges remain: (1) How will we determine the azimuth and elevation angle from the 1-D an-

gle reported by MUSIC? (2) How can we ensure fine-grained tracking by using the incoming signals from a modest number of transmitters? (3) While 2D hand position may be obtained by considering the azimuth and elevation of an AoA, how can we track the hand's depth to enable 3D motion tracking? (4) The incoming signal from a transmitter can change if the transmitter is mobile or the wireless environment is changing quickly. How can we enable robust motion tracking under such AoA changes? The next section answers these questions and presents our system called *WiDraw* that enables fine-grained hand trajectory tracking using AoA values of wireless signals.

3. HAND TRACKING ALGORITHMS

This section presents our solutions that enable tracking the user's hand near the WiFi receiver. We begin by explaining how the azimuth and elevation of an AoA can be computed. Thereafter, we explain how we track the hand's motion using the azimuth and elevation values.

3.1 Azimuth and elevation estimation

To calculate the azimuth and elevation values from MUSIC's 1-D AoA output, we refer to Equation 1. Observe in Figure 2(b) that, if the user rotates the linear antenna array over the Z axis (as shown in Figure 6), the azimuth of a given AoA will gradually increase until the signal incidents the laptop along its antenna array, when the azimuth becomes 90^o. At that moment, the 1-D AoA value reported by MUSIC (θ in Figure 2(b)) will attain its maximum value and according to Equation 1 it will essentially be equal to the (90^o - elevation angle). Therefore we ask the user to turn the laptop around its Z-axis as shown in Figure 6 and collect MUSIC's pseudospectrum outputs. From the pseudospectrums, for each 1-D AoA, we compute the maximum value it attains and use it to compute the elevation. Once we know the elevation, the azimuth of the AoA is estimated using Equation 1.

A pertinent question is how frequently will the user have to turn the laptop for motion tracking? Observe that a recalibration is only required when the direction of incoming signals from a given transmitter changes completely. This happens when an existing dominant incoming signal direction is blocked and thus, another incoming signal appears with a different AoA value. We will explain in Section 4.3 how we determine such changes and automatically calculate the new azimuth and elevation values, without any user initiated calibration.

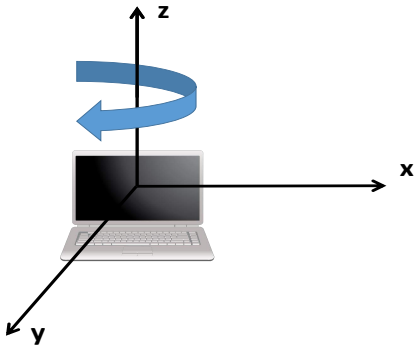

Figure 6: User turns the laptop around its Z-axis to help *WiDraw* compute the per AoA azimuth and elevation.

3.2 Hand trajectory tracking

During the hand's motion, if an AoA's signal strength attains a minimum, we can calculate the instantaneous spatial coordinates of the hand by applying the AoA's azimuth and elevation values in Equation 2. However, Equation 2 can compute the horizontal and vertical coordinates of the hand only if the hand's depth from the receiver is known. Therefore, we begin by discussing how the hand's depth can be estimated.

Determining hand's depth

To determine the hand's initial depth, we track the average of the RSSI values of all incoming signals from the neighboring transmitters. Figure 7(a) plots the drop in the average RSSI value when the user's hand is placed at varying depths from the receiver. We find that, on average, the drop in the average RSSI is higher if the user's hand is placed closer to the receiver. This happens because, when the hand is closer to the receiver's antenna, a larger number of incoming signals from different transmitters are blocked. Therefore, we calculate the drop in the average RSSI value when the user's hand first appears near the receiver. Thereafter, we calculate the approximate initial depth of the hand using the relationship established in Figure 7(a).

The depth of the user's hand will not change in case the user is drawing on a plane parallel to the receiver's antenna. However, if the user's hand moves in all 3 dimensions, the depth value will not remain constant. By tracking the changes in the average RSSI value across all transmitters, it is possible to approximately determine the depth changes as well. We try to further improve the accuracy of depth estimation by analyzing the time difference between consecutive AoA minima. Our intuition is that, if the hand is moving towards the receiver, the time difference value between consecutive AoAs will be smaller than if the hand is moving away from the receiver. By analyzing the time difference values, it is possible to geometrically find the change in the hand's depth. E.g., in Figure 7(b) let the user's hand pass through 3 consecutive AoAs - AoA_0, AoA_1, and AoA_2 at times t_0, t_1, and t_2, respectively. The depth of the hand changes by h_1 units between t_0 and t_1 and by h_2 units between t_0 and t_2. Since we know the initial depth of the hand at time t_0, we can compute the coordinates of the points A, B, C, which capture the trajectory of the hand if its depth remained constant. If d_1, d_2 are the distances between these coordinates (as shown in 7(b)), geometrically we can obtain the change in depth h_2 at time t_2 as:

$$h_2 = \frac{(d_1 + d_2) * \alpha - d_1}{(\tan\theta_2{}^{-1} - \tan\theta_1{}^{-1}) * \alpha} \quad (3)$$

in which $\alpha = (t_1 - t_0)/(t_2 - t_0)$. By utilizing the above equation, we can determine the change in depth of the hand's trajectory. This, when combined with the hand's initial depth, allows us to track the hand's absolute depth.

Determining horizontal and vertical coordinates

The spatial coordinates computed from an AoA when it attains a minimum may not represent the exact coordinates of the user's hand. This is because, if the user's hand does not occlude the AoA completely but is placed near its trajectory, the AoA's signal strength will still drop. Figure 7(c) shows the drop in signal strength with increasing proximity of the hand

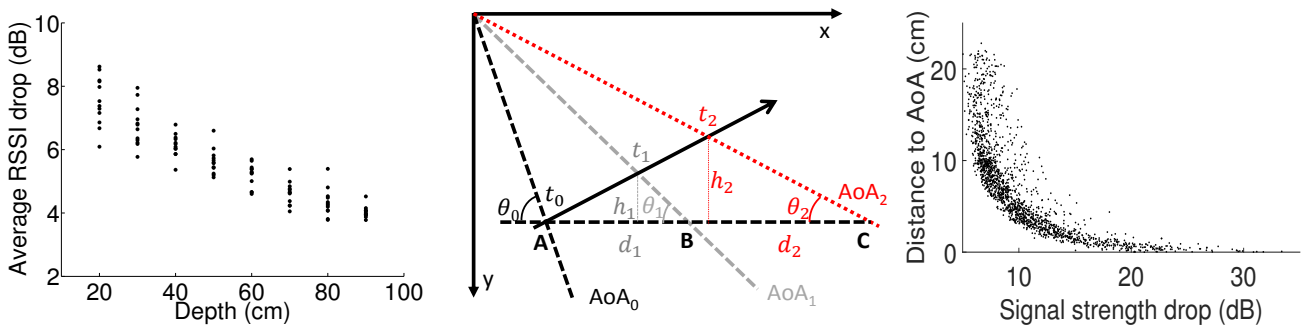

Figure 7: (a) Average RSSI drop across all AoAs vs. depth of the hand from the *WiDraw* receiver. 25 locations in the same 2x2 square feet area were picked uniformly at each depth value. (b) Estimating depth changes by utilizing the time duration between AoA minima. The user's hand crossed 3 AoAs (AoA_0, AoA_1, and AoA_2) in a 3D movement. The trajectories of the hand movement and 3 AoAs were projected to the 2D X-Y plane. (c) The drop in signal strength of an AoA is proportional to the hand's distance from the AoA.

to the AoA's trajectory. The drop in signal strength can therefore suggest the approximate distance of the hand from the coordinates computed from the AoA's azimuth and elevation values. E.g., consider AoA_0 in Figure 8. If the drop in signal strength the moment it attains its minimum value is 8.5 dB, the hand's actual location can be anywhere on the perimeter of a circle with a radius of 5 cm. Further, the trajectory of the user's hand at that moment is also tangential to the same circle.

We determine the location of the user's hand on the circle by analyzing the trend (increasing or decreasing) of the neighboring angles' signal strength values at the moment when AoA_0 attains a minimum. If the signal strength of a neighboring angle is increasing, then the hand's direction is away from it and vice versa. E.g., in Figure 8, the hand is moving towards AoA_1 and AoA_3 but is moving away from AoA_2. Consider the unit vectors along the tangents in Figure 8 which pass through the coordinates obtained from the neighboring angles. The direction of a particular unit vector is towards the angle if the angle's signal strength is decreasing, and away from the angle if it is increasing. The hand's direction is closer to the direction of a unit vector if the corresponding angle's signal strength is decreasing faster. Therefore, we aggregate these unit vectors by computing their weighted average based on the rate of change of the corresponding angle's signal strength. Mathematically, if the rate of change in the signal strength of the neighboring angle AoA_i is v_i, and the corresponding unit vector is $\hat{u_i}$, the weighted average ($\hat{U}$) is

$$\hat{U} = \frac{\sum_{i=1}^{n} |v_i| * \hat{u_i}}{\sum_{i=1}^{n} |v_i|} \quad (4)$$

The direction of the aggregate vector $\hat{U}$ provides us the estimate of the hand's direction. Observe in Figure 8, there can be only 2 tangents to the circle whose direction matches with that of $\hat{U}$. The hand's coordinate can be either of the points where these two tangents meet the circle (A and B in Figure 8). To select one of the two points, we (i) use the coordinates of A or B to estimate signal strength drop for each AoA based on the relationship established in Figure 7(c) and (ii) calculate the actual signal strength drop for each AoA by

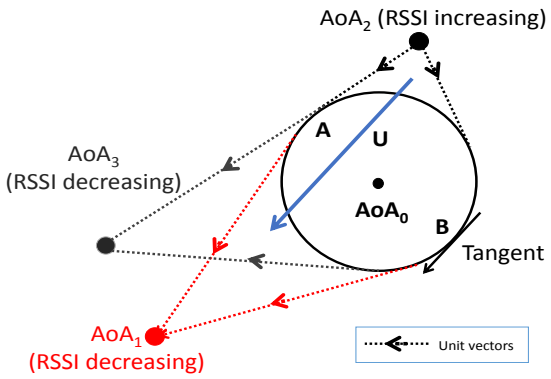

Figure 8: Determining spatial coordinates for the user's hand when AoA_0 attains a minimum, using the signal strength trend of neighboring angles.

comparing the currently measured signal strength value with the initial one. Then, we can break the tie between A and B by choosing the point which yields a more accurate estimate of signal strength drop for most AoAs.

3.3 Dealing with large environmental variations

We found in Section 2.4 that the changes in signal strength due to random environmental related fluctuations have a different structure than the changes due to the user's hand movement. Such random fluctuations can be ignored by applying filtering and by ignoring any signal strength drop which is smaller than 3dB (Figure 5(b)). Large and gradual drop in the signal strength of an AoA due to environmental variations can still occur. If large environmental variations affect an AoA near the hand's trajectory, this will reflect in our trajectory tracking error. However, if such an AoA is far away from the hand's trajectory, it needs to be detected as an outlier. Further if the time interval between consecutive AoA minima is too long, then they were probably a result of random environmental fluctuation and not due to the hand's motion. Formally, if θ_{i-1} and θ_i are two consecutive AoAs which demonstrated a large drop in signal strengths at time t_{i-1} and t_i,

and resulting into 2D coordinates C_{i-1} and C_i respectively, we declare θ_i as an outlier if

$$\frac{|\hat{C_i} - \hat{C_{i-1}}|}{t_i - t_{i-1}} > \tau_m \quad (5)$$

where τ_m is the maximum expected speed of the user's hand, which is fixed at 3 feet/second in our implementation. We also declare both θ_{i-1} and θ_i as outliers if the time duration between them is more than a threshold (T):

$$t_i - t_{i-1} > T \quad (6)$$

To determine the accuracy of identifying the AoAs whose signal strengths change due to the hand's motion and not because of large environmental fluctuations, we collected 2 sets of traces over a duration of 6 hours. The first trace consists of AoA data when the user draws in front of the receiver's antenna, while in the second trace she is sitting idle in front of the receiver. Table 1 shows the accuracy and false positives of identifying the relevant AoAs computed from the above 2 sets of traces. If we consider AoAs that undergo a significant drop in signal strength within a considerably long duration (T in Equation 6) as valid, false positives expectedly increase, and so does accuracy. However, as long as we reject AoAs that undergo minima beyond 1.5 seconds, it is possible to maintain low false positive rate with a slight decrease in accuracy. We further note that while using a threshold of 1.5 seconds, 13 short trajectories were falsely recorded within a duration of 6 hours, implying that our false positive rate is less than 0.04 events/minute.

4. DESIGN AND IMPLEMENTATION

In this section, we present *WiDraw*'s detailed design.

4.1 Setup and implementation details

In our solution, we equip the laptop with a wireless network card equipped with Atheros 9590 chipset. The network card is attached to 3 antennas and is able to extract the Channel State Information (CSI) from all antennas on receiving a wireless frame. The reported CSI is a matrix containing one complex number per subcarrier and per receive antenna at the receiver. During the calibration period, the Atheros card is set in monitor mode to capture the MAC addresses of neighboring 802.11 devices – both APs and clients. If the neighboring device is a client, *WiDraw* may need to probe it to obtain the CSI information from the received ACK. To keep the probing overhead low, the *WiDraw* receiver uses link layer NULL frames to probe a client device. Note that neighboring devices will respond to the NULL frame with an ACK, irrespective of their connection state. Thereafter, *WiDraw* uses the CSI values from the ACK to compute and analyze the AoA values.

Time duration threshold (sec)	Accuracy	False positives
0.5	84.8%	0.02%
1.0	87.8%	0.03%
1.5	89.5%	0.06%
2.0	91%	0.11%

Table 1: Accuracy & false positive rate of identifying the relevant AoAs whose signal strength changes due to hand occlusions and not due to environmental fluctuations.

Implementation details *WiDraw* is implemented as a combination of driver modifications and user-space programs. The WiFi driver collects CSI and RSSI values of incoming signals and sends it to a userspace program written in C. The userspace program calculates the 1-D AoAs using MUSIC. It also calculates the azimuth and elevation values for each AoA and reports these values along with the signal strength changes over time to another module written in MATLAB, which uses this information to track the hand's trajectory.

4.2 Transmitter selection

Apart from APs, *WiDraw* should only utilize stationary client devices for AoA estimation since, if the device is mobile, its AoA values will not be stable. To determine if a device is actually static, we adopt the algorithm as studied in [13]. Briefly, we compute the correlation of the CSI values over time, which captures the similarity between a current and a past CSI value. Low correlation values imply rapid changes in the wireless channel between the client device and the *WiDraw* receiver, due to mobility or environmental variations. Therefore, in our implementation, we avoid clients whose CSI correlation values are lower than 0.9. Further, we observe that the user will mostly draw in front of the laptop's screen. Therefore, we optimize by selecting only those transmitters whose AoAs fall within the same region.

4.3 Azimuth and elevation estimation after user initiated calibration

The transmitter selection procedure runs during the user initiated calibration phase when the azimuth and elevation of the AoAs are identified. After the calibration is completed, the AoA value of a chosen client device may change. Further, new client devices whose AoAs can be suitably used for trajectory tracking may appear in the neighborhood. It is impractical to ask the user to calibrate whenever such changes happen. To detect the azimuth and elevation values of these unknown AoAs without calibration, we track their signal strength while a user is drawing in the air. If an unknown AoA's signal strength demonstrates a minimum at a given moment, we choose the coordinates obtained by the AoAs whose minima occurred right before and after the unknown AoA. Thereafter, we compute the coordinates of the user's hand during the unknown AoA's minimum as a weighted average of the two chosen coordinates. Once the coordinates are known we use Equation 2 to compute the azimuth and elevation of the unknown AoA. Thereafter, the new AoA can be used to track the user's hand.

4.4 System architecture

Figure 9 shows the overall architecture of *WiDraw*. During the calibration phase, the laptop's WiFi card is set in monitor mode to capture MAC addresses of nearby 802.11 devices and select the initial set of transmitters (Section 4.1). During the same phase, the receiver uses the CSI of incoming signals and MUSIC to estimate their 1-D AoA values (Section 2.1). Then, the user is asked to rotate the laptop and the system calculates the azimuth and elevation of each AoA (Section 3.1).

During normal operation, the *WiDraw* receiver periodically collects RSSI and CSI from incoming signals. Then, it uses CSI correlation values over time (Section 4.2) and filtering (Section 3.3) to filter out signals from mobile clients or those affected by large environmental variations. For stable AoAs, *WiDraw* processes the RSSI value and obtains an estimate of

the hand's depth (Section 3.2). If a minimum signal strength value of an AoA is detected, the depth estimation is refined using Equation 3. If the azimuth and elevation of the AoA are known, we use the algorithm in Section 3.2 to estimate the hand's coordinates. Otherwise, for unknown AoAs, the hand's coordinates are estimated as the weighted average of the coordinates of two known neighboring AoAs (Section 4.3).

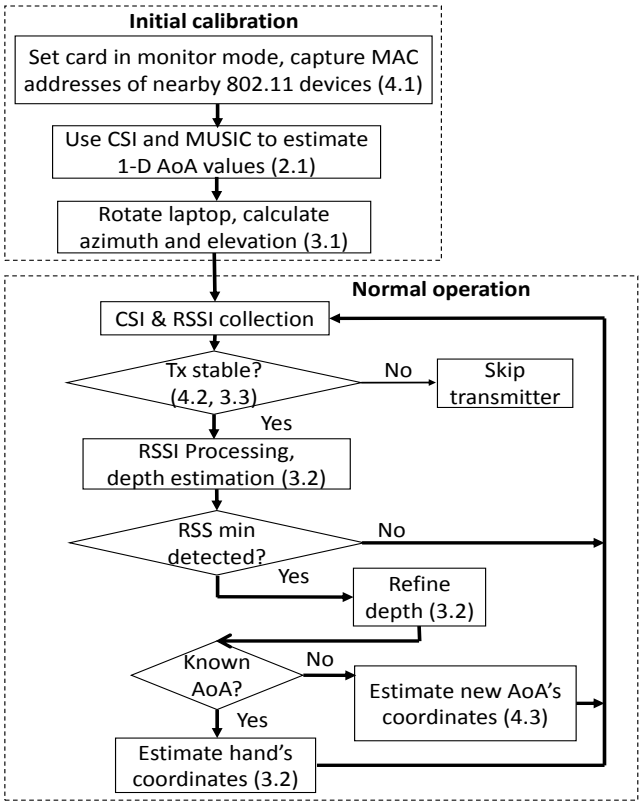

Figure 9: *WiDraw* Architecture.

5. EVALUATION

In this section, we evaluate *WiDraw*'s trajectory tracking accuracy when the user draws 2D and 3D lines, curves, and English alphabetical characters. Unless otherwise stated, the user sits in front of the laptop and starts drawing at a location within 2 feet distance from the laptop's antenna, at an approximate speed of 1.5 feet/second. We evaluate *WiDraw* in a typical office building during normal business hours as well as a busy cafeteria. In the default case, the laptop computes AoA values from 30 neighboring transmitters, including both APs and client devices. For each transmitter, *WiDraw* computes the AoA values 20 times/second.

Ground truth collection: We compute the ground truth of the hand's trajectory by collocating a Leap Motion controller [2] with the laptop's WiFi antenna. Leap Motion reports the 3D location of the palm in millimeters w.r.t. the Leap Motion controller as the origin, which we use as the ground truth coordinates of the user's hand. *WiDraw*'s tracking error is computed by calculating the point-by-point position difference between the ground truth trajectory and *WiDraw*'s estimated trajectory.

5.1 2D Tracking Evaluation

We begin by evaluating the accuracy of tracking a total number of 8448 2D movements including line segments and circular curves on a 2D plane parallel to the laptop's screen. Figure 10(a) shows the distribution of tracking errors for both lines and curves. The error increases with increasing depth from the laptop's antenna (Figure 10(b)). The impact of increasing depth is twofold: first, the distance between consecutive coordinates obtained from the AoAs becomes larger; second, the drop in signal strength at a particular AoA due to hand occlusion becomes weaker and may be mistakenly attributed to environmental noise. Nonetheless, at a depth of 2 feet (60 cm) from the laptop, *WiDraw*'s median tracking error is 5.4 cm. The median tracking error does not reduce significantly if the neighboring APs or client devices are probed with a greater number of probe packets every second (Table 2). *WiDraw*'s tracking error also does not accumulate over time. Table 2 shows that *WiDraw* can track reasonably long trajectories (up to 300 seconds) with only a slight increase in estimation error. *WiDraw*'s tracking error however depends on the number of neighboring transmitters used for AoA estimation (Figure 10(c)). Nonetheless, even by using only the 12 neighboring transmitters, *WiDraw* can track the user's hand with a median error lower than 6.4 cm.

Packets/sec	Drawing time (seconds)				
	10	30	60	120	300
100	4.41	4.76	4.77	4.81	4.82
50	4.42	5.01	5.05	5.06	5.05
25	4.83	5.01	5.11	5.09	5.09
10	5.22	5.34	5.71	5.76	5.76

Table 2: Median tracking error (cm) for varying probing frequency and drawing time.

5.2 3D Tracking Evaluation

We evaluate the accuracy of *WiDraw* in distinguishing between 2D and 3D movements and tracking the 3D movements. The user draws a total of 1560 straight lines or curves while moving the hand either towards or away from the laptop. Table 3 shows the accuracy and false positives of distinguishing between 2D and 3D movements by utilizing the change in the average RSSI value, as described in Section 3.2. Once we determine that the user is drawing in all 3 dimensions, we start tracking the changes in the hand's depth by using the algorithm specified in section 3.2. Figure 11(a) shows that while *WiDraw* can often estimate the depth change with less than 10% error, there are occasions when the estimated depth change does not conform with the actual depth change, resulting into tracking errors. Figure 11(b) shows the distribution of the 3D trajectory estimation errors. The median tracking error (8.6 cm) is higher than in the case of 2D trajectories because of an additional error introduced by changing depth values. Therefore, we find that the 3D tracking error is proportional to the depth estimation error (Figure 11(c)).

5.3 Tracking English Characters

Next, we evaluate if *WiDraw* can effectively track the trajectory of English characters drawn by the user. English characters are more difficult to track than simple drawings because they consist of multiple sharp turns, segments, etc. In our study, we used 75,880 characters drawn by a total of 3 users.

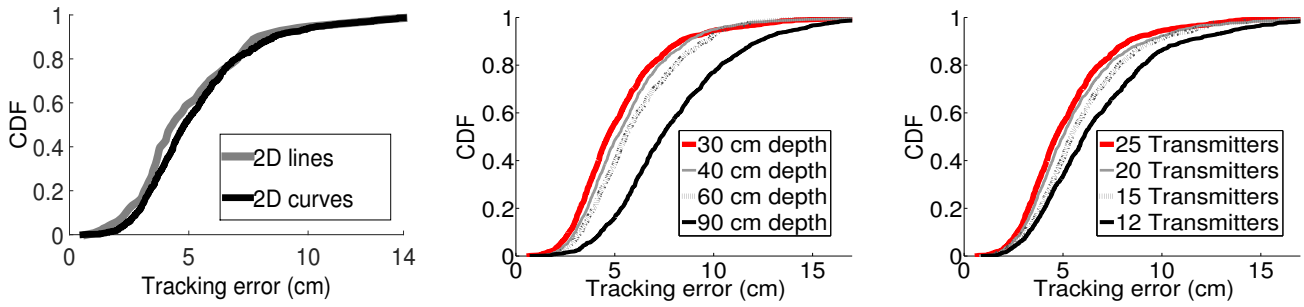

Figure 10: (a) CDF of tracking error for 2D lines and curves drawn by the user. Tracking error increases with (b) depth of the user's hand from the receiver, (c) decreasing number of neighboring transmitters.

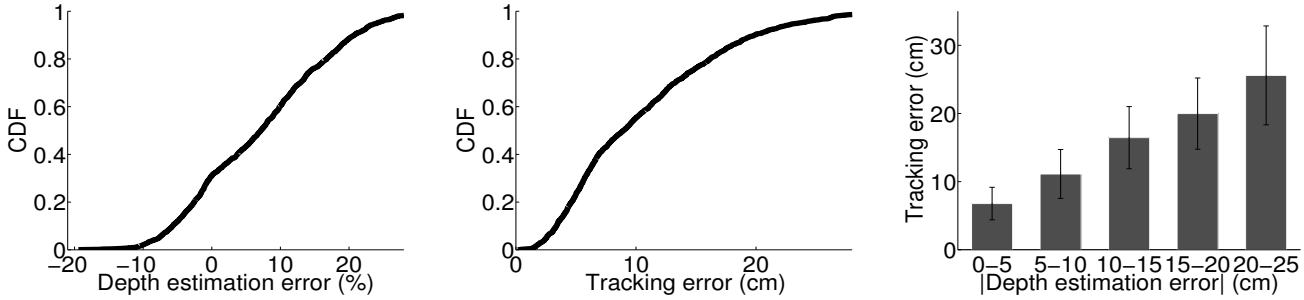

Figure 11: (a) CDF of depth change estimation error for 3D trajectories. Percentage of error calculated as (ground truth depth change - estimated depth change)/(ground truth depth change). Ground truth depth change varies between 1 and 3 feet. (b) CDF of tracking error for 3D lines. (c) Tracking error increases with increasing error in depth estimation.

Ground Truth	Detection result(%)	
	2D	3D
2D	98.36	1.64
3D	2.81	97.19

Table 3: Distinguishing between 2D and 3D trajectories.

We designed experiments to determine the dimensions of the 3D area, near the receiver, within which *WiDraw* can effectively identify the letters drawn by the user. We asked the users to draw letters of different sizes naturally, at varying distance from the receiver. Figures 12(a) and 12(b) show the distribution of trajectory tracking error when the users draw characters at increasing depth and horizontal distance from the receiver respectively. We observe that, for a depth and horizontal distance smaller than 2 feet, the median tracking error remains lower than 5.2 cm.

Figure 12(c) shows that the tracking error increases with the letter size, since, for large sizes, some segments of the trajectory will be farther away from the receiver's antenna. However, notice that the tracking error does not fully capture the *WiDraw*'s capability of identifying letters. Even with large tracking errors, *WiDraw*'s estimated trajectory can be successfully recognized by a simple handwriting recognition app if there are enough AoA minima, as we evaluate in the next section. This happens because *WiDraw*'s errors are not independent random errors. Most of the errors are due to depth estimation errors or lack of motion information between two AoA minima. Therefore, although the tracking error is a few centimeters, the estimated trajectory is either an enlarged or squeezed version of the original letter. Similarly, a very small tracking error may not yield high recognition accuracy if there are not enough AoA minima to capture every segment of the trajectory. As an example, from the results in Figure 12(c) we find that for a reasonable letter size of 30 cm, *WiDraw*'s median tracking accuracy is approximately 4.1 cm. In the next section, we show that even with such errors, a simple handwriting recognition app can successfully identify the letters in up to 98% of the cases.

5.4 Evaluation in different environments

The above results demonstrate *WiDraw*'s performance in an office environment. In this section, we evaluate *WiDraw*'s performance in a busy cafeteria and in NLOS environments where the user's hand is not visible to the receiver. In the cafeteria experiment, a user draws several curves and letters at a distance of 1 foot from the laptop screen. Figure 13(a) plots *WiDraw*'s tracking error when dealing with environmental noise and automatically adjusting for changes in AoA values. The error is higher than in the experiments in the office building because of the environmental noise. Recall that *WiDraw* computes the azimuth and elevation values of the AoA by asking the user to turn the receiver around its Z-axis. Figure 13(b) shows the median tracking error over time after a user calibration. Figure 13(b) also shows the tracking error over time in an NLOS scenario when the *WiDraw* receiver is placed in a bag, while the user is drawing lines in the air at a distance of 1 foot from it. We find that after 1 hour since calibration, the median tracking error in both the LOS and the NLOS scenario is within 6 cm.

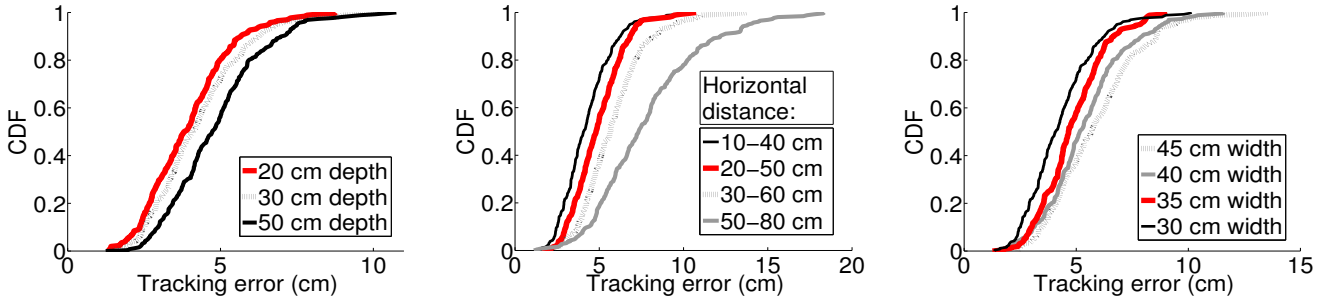

Figure 12: CDF of trajectory estimation error for upper and lower case English letters with (a) increasing depth (horizontal distance: 10-40 cm, letter width: 30 cm) and (b) horizontal distance from the receiver's antenna (depth: 30 cm, letter width: 30 cm). (c) Tracking error increases with increasing width of the letters (depth: 30 cm, horizontal distance: 10-40 cm).

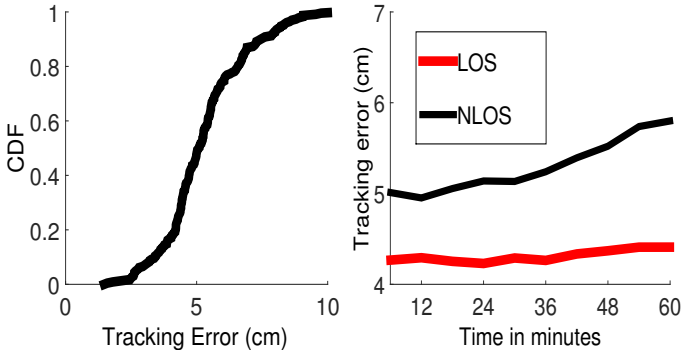

Figure 13: (a) CDF of trajectory estimation error in a busy cafeteria. (b) Estimation error over time in both LOS and NLOS environments.

6. USING *WiDraw* FOR TEXT INPUTS

In this section, we show how *WiDraw* can be used to allow the user to reliably input texts to a computer by writing in the air. Similar to the approach in [7], we interface *WiDraw* with the MyScript Stylus app on Android [14]. As the user writes in the air, we feed *WiDraw*'s estimated trajectory into MyScript and we use its handwriting recognition functionality to interpret the text written by the user. Using MyScript's output, we evaluate the character and word recognition accuracy of *WiDraw*. We begin by studying the effect of various constraints on letter recognition accuracy.

6.1 Letter Recognition Accuracy

Table 4 shows *WiDraw*'s letter recognition accuracy for both capital and small letters drawn by the user. The accuracy drops with increasing depth and horizontal distance of the user's hand from the receiver's antenna. *WiDraw*'s letter recognition accuracy is less than 80% beyond 2 feet from the receiver. The above results suggest that for good results the user should draw the characters within a $3.3 \times 2 \times 2$ cubic feet area in front of the laptop's antenna. The letter recognition accuracy also decreases with decreasing character sizes (Figure 14), but the accuracy is quite high for sizes greater than 30 cm. We find that even if the median trajectory tracking error for letters is around 5 cm (as demonstrated in section 5.3), characters of reasonable sizes can still be recognized because most of the errors are due to transformation and distortion in the shape of the trajectory rather than independent positioning errors.

Horizontal range (cm)	Depth (cm)			
	30	40	50	60
0-30	0.98/0.97	0.95/0.87	0.95/0.82	0.91/0.80
10-40	0.98/0.96	0.96/0.94	0.92/0.87	0.89/0.79
20-50	0.96/0.92	0.90/0.83	0.89/0.80	0.87/0.77
30-60	0.88/0.89	0.85/0.78	0.85/0.79	0.76/0.73

Table 4: Letter recognition accuracy for uppercase/lowercase English letters (letter width 30 cm).

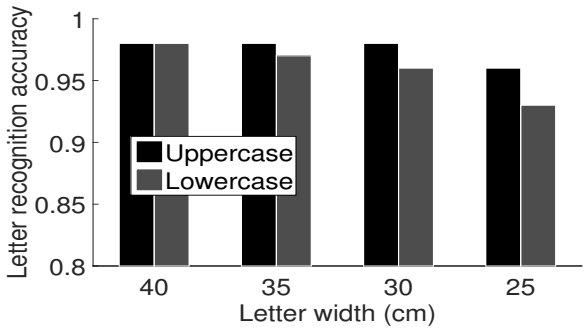

Figure 14: Letter recognition accuracy decreases with decreasing width of the letters. Distance from Rx: 1 foot.

6.2 Writing Words using *WiDraw*

WiDraw allows users to write cursive words in the air. The average size of a user's handwritten letter that is supported by *WiDraw* is around 30 cm. Further, the width of the horizontal plane on which the user can write is limited to 100 cm. Therefore, while writing words, we allow the users to write up to 3 characters contiguously. If a word consists of more than 3 characters, the user needs to start drawing the 4th character from the beginning of the line. Figure 15(a) shows *WiDraw*'s estimated trajectory when the user writes "*widraw*" in the air. Notice that the estimated trajectory overlaps "wid" with "raw" and includes a horizontal line between the 3rd ("d") and the 4th ("r") character and hence it is hard to recognize. *WiDraw*

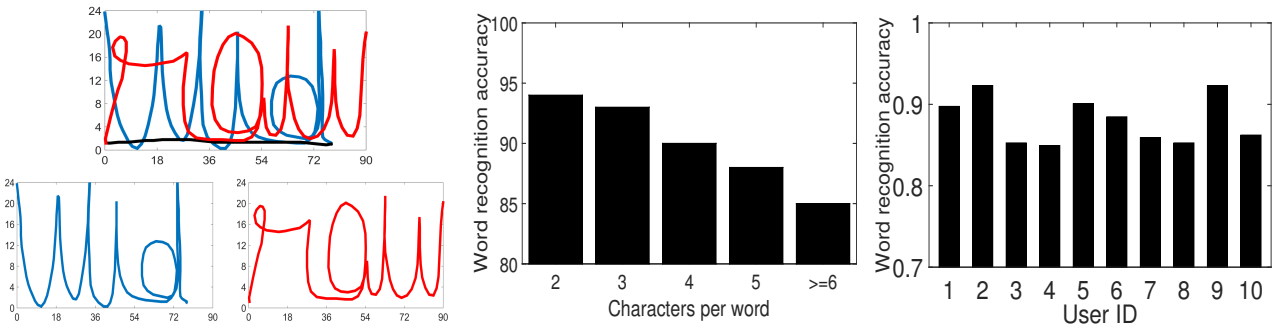

Figure 15: (a) User starts every 4th character in a newline, *WiDraw* segments the drawing to make the letters contiguous. (b) Word recognition accuracy reduces with increasing number of letters per word. (c) Word recognition accuracy for 10 different users when the users write continuous sentences.

determines this horizontal line, removes it from the trace, and then places every 3 letters in sequence (Figure 15(a)). The horizontal line separating two groups of 3 letters will be close to 90 cm since each letter's approximate width is 30 cm. Therefore, if the user is drawing words, *WiDraw* determines any horizontal straight line which is more than 2 letters wide (> 60 cm) and then segments the trace into separate halves based on the starting and ending coordinates of the straight line.

We evaluate *WiDraw*'s word recognition accuracy by estimating the trajectories of 1000 words drawn by 3 users at a distance of 1 foot from the receiver, and thereafter feeding these trajectories into the MyScript Stylus app. Figure 15(b) shows *WiDraw*'s word recognition accuracy as a function of the number of letters in the word. Expectedly, as the word gets longer, it is more difficult to recognize it correctly. However, even by using a simple handwriting recognition software such as MyScript, the average word recognition accuracy is more than 91%. Of course the accuracy can be improved by using advanced techniques in natural language processing [15].

6.3 Writing Sentences using *WiDraw*

To enable the user to write sentences using *WiDraw* we ask the users to pause for a moment (typically 1 second) between consecutive words. Since the user pauses between words, there will be a little change in the signal strength of different AoAs in between two words. Therefore, to segment words in a sentence, we track the rate of change in signal strength of different AoAs. If the signal strength of a threshold percentage of AoAs does not change for more than 500ms, we classify it as a pause and segment the words accordingly. The accuracy of pause detection reduces slightly if we increase the threshold of the percentage of AoAs. However, false positives may imply wrongly breaking up a single word into multiple words. In our implementation, we chose a threshold of 92% since it results into no false positives and achieves a high word segmentation accuracy of 96%.

An undetected pause may not necessarily mean an error in word recognition. This is because MyScript can sometimes detect if a sequence of characters actually consist of multiple words. To evaluate the accuracy of writing multiple words we asked 10 users to write a passage of 100 words continuously in the air, with 3 characters per line and a pause between two words. Figure 15(c) shows that even with a simple Android handwriting recognition software such as MyScript, *WiDraw*'s word recognition accuracy for continuous writing is higher than 85% across all users. Note that, for the evaluation of RF-IDraw [7], the system closest to ours, the authors manually segmented the user's writing into words. As pointed out in [7], standard segmentation methods from natural language processing [16] can be used to further improve the word segmentation accuracy. Overall, the result in this section implies that *WiDraw* can be used to input different kind of texts to the mobile device.

7. RELATED WORK

Gesture recognition and hand motion tracking related research can be classified in three types.

Camera-based Vision-based gesture recognition has been a well-researched topic for over two decades [17, 18]. Recent works use depth cameras (e.g., [1]) or infrared cameras (e.g., [2, 19]) to enable in-air 3D human computer interactions. Despite their popularity, these approaches require a dedicated hardware setup and LOS to the user.

Motion sensor-based Some recent works rely on motion sensors available in today's smartphones to perform gesture recognition, or allow the user to write in the air (e.g., [20, 21]). Others make use of such sensors on wearables such as smartwatches [22], armbands [23], wristbands [24], and rings [25, 26]. In contrast to *WiDraw*, all motion-based systems require an external device such as a smartphone or a wearable. Moreover, most of these approaches rely on machine learning to achieve fine-grained gesture recognition.

RF-based *WiDraw* belongs to a recent class of systems which employ RF signals to enable a user to interact with their environment in both LOS and NLOS scenarios [3–8, 27, 28]. Most of these works [3–6, 8, 27] require the use of custom wireless hardware and/or signal processing capabilities unavailable on off-the-shelf devices. In addition, several solutions [3–5,8] require a priori learning of the effect of hand motions on wireless signal patterns, which allows them to recognize only a pre-defined set of gestures; others [6, 27] focus on human motion tracking rather than fine-grained gesture recognition and can only identify coarse-grained gestures (e.g., a pointing gesture).

Wi-Fi Gestures [28] and WiGest [29] are the only two works to our best knowledge that perform WiFi-based gesture recognition using RSSI and CSI information from off-the-shelf devices. However, they also rely on a priori learning, and as a

result, they can only classify a few simple gestures. Lastly, RF-IDraw [7] is the first RF-based system that can accurately track the trajectory of the hand, enabling a virtual touch screen based on RF signals. While RF-IDraw can achieve good tracking accuracy, it requires the user to hold an RFID transmitter, similar to motion sensor-based solutions.

WiDraw's contribution: In contrast to previous solutions, *WiDraw* is the first hands-free motion tracking solution that can be enabled on existing mobile devices using only a software patch, combining the desired features of the above two approaches. Similar to Wi-Fi Gestures and WiGest, it uses information readily available from commodity devices, without requiring the user to hold any device; and similar to RF-IDraw, it can identify an arbitrary number of fine-grained gestures without any requirements of a priori learning.

8. DISCUSSION AND LIMITATIONS

Gesture recognition vs. hand motion tracking: In-the-air user interfaces can be divided into two classes. The first class is based on defining a limited set of gestures and using machine learning to learn patterns and classify gestures into the learned categories [3–5, 8, 28]. The second class includes interfaces enabled by systems such as RF-IDraw [7] and *WiDraw*. These interfaces require no priori learning and can track an arbitrary set of hand motions, enabling a much richer set of applications.

Distance from the receiver's antenna: Our prototype achieves satisfactory tracking accuracy if the hand is within 2 feet from the receiver's antenna. The error at larger distances can be reduced if the receiver is equipped with a larger number of antennas. One way to achieve this is to equip the receiver with multiple wireless cards [10]. Future mobile devices may also be equipped with more antennas – the 802.11ac standard [30] provides support for up to 4 antennas on client devices.

Number of transmitters: *WiDraw* requires at least a dozen of transmitters in order to track the user's hand with high accuracy. We believe this number is not a challenge in today's enterprise and home environments in urban areas. A recent study [31] reports a median number of 17 and a maximum number of 78 neighboring APs in dense urban environments. This number can be further reduced if the receiver is equipped with a larger number of antennas, which is the current trend as mentioned above. Alternatively, if the number of APs is not sufficient, WiDraw may also use nearby clients. We further emphasize that, similar to other RF-based motion tracking systems, *WiDraw* does not require LOS between the transmitter and the receiver.

Probing overhead: If the number of neighboring APs is not sufficient, or if we need a higher frequency of CSI sampling, WiDraw may also probe APs or nearby clients. We believe that this probing overhead can be kept low. For example, Table 2 shows that the tracking error does not increase significantly when the probing frequency is reduced from 25 packets/second to 10 packets/second (a typical beacon transmission rate). We leave a detailed study of the impact of probing on the performance of other ongoing WiFi transmissions as future work.

Calibration: *WiDraw* asks the user to turn the laptop for azimuth and elevation computation during setup. An alternative approach we plan to investigate as part of our future work is to ask the user to perform a set of pre-defined gestures (similar to Leap Motion [2]) and compute the azimuth and elevation by observing the angle minima.

Depth estimation error: The 3D tracking error of *WiDraw* is larger than the 2D error primarily due to the difficulty in accurately tracking depth changes; improving the depth estimation error is part of our future work. However, note that, while accurate 2D motion tracking is required for enabling a virtual touch screen, several applications often need only a very small set of 3D gestures (e.g., push, pull in gaming), where distinguishing between 2D and 3D motion (which *WiDraw* achieves with 97% accuracy) is more critical than accurately tracking the depth change.

9. CONCLUSION

This paper introduced *WiDraw*, the first hand motion tracking system in both LOS and NLOS scenarios using commodity WiFi cards. *WiDraw* harnesses the AoA values of incoming wireless signals utilizing the intuition that whenever the user's hand occludes a signal coming from a certain direction, the signal strength along that direction will experience a drop. Unlike prior solutions, *WiDraw* can be enabled on today's mobile devices using only a software patch, without requiring the use of any wearable or any dedicated hardware setup. Our prototype using commodity wireless cards can track the user's hand with less than 5 cm error on average. We also use WiDraw to implement an in-air handwriting application that achieves an average word recognition accuracy of 91%. We believe that *WiDraw* can open up a whole new class of applications in human-computer interaction.

10. ACKNOWLEDGMENT

We sincerely thank the anonymous shepherd and MobiCom reviewers for their insightful feedback.

11. REFERENCES

[1] Microsoft kinect. http://www.microsoft.com/en-us/kinectforwindows/.

[2] Leap motion, inc. leap motion: Mac & pc gesture controller for game, design and more. https://www.leapmotion.com/.

[3] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of ACM Mobicom*, 2013.

[4] Fadel Adib and Dina Katabi. See through walls with WiFi! In *Proceedings of ACM SIGCOMM*, 2013.

[5] Bryce Kellogg, Vamsi Talla, and Shyam Gollakota. Bringing gesture recognition to all devices. In *Proceedings of USENIX NSDI*, 2014.

[6] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 3D tracking via body radio reflections. In *Proceedings of USENIX NSDI*, 2014.

[7] Jue Wang, Deepak Vasisht, and Dina Katabi. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proceedings of ACM SIGCOMM*, 2014.

[8] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proceedings of ACM Ubicomp*, 2014.

[9] Ralph O Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34(3):276–280, 1986.

[10] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. Phaser: enabling phased array signal processing on commodity wifi access points. In *Proceedings of ACM Mobicom*, 2014.

[11] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of ACM MobiSys*, 2013.

[12] Kainam Thomas Wong and Michael D Zoltowski. Root-music-based azimuth-elevation angle-of-arrival estimation with uniformly spaced but arbitrarily oriented velocity hydrophones. *Signal Processing, IEEE Transactions on*, 47(12):3250–3260, 1999.

[13] Li Sun, Souvik Sen, and Dimitrios Koutsonikolas. Bringing mobility-awareness to WLANs using PHY layer information. In *Proceedings of the ACM CoNEXT*, 2014.

[14] Myscript. Myscript stylus app. http://myscript.com/.

[15] Wenwei Wang, Anja Brakensiek, and Gerhard Rigoll. Combination of multiple classifiers for handwritten word recognition. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 117–122. IEEE, 2002.

[16] Réjean Plamondon and Sargur N Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[17] James M Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Proceedings of the European Conference on Computer Vision*, 1994.

[18] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, 1995.

[19] Nintendo wii. http://www.nintendo.com/wiiu.

[20] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. Using mobile phones to write in air. In *Proceedings of ACM MobiSys*, 2011.

[21] Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, and Junehwa Song. E-Gesture: A Collaborative Architecture for Energy-efficient Gesture Recognition with Hand-worn Sensor and Mobile Devices. In *Proceedings of ACM SenSys*, 2011.

[22] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proceedings of ACM HotMobile*, 2015.

[23] Myo: Gesture control armband. https://www.thalmic.com/en/myo/.

[24] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. RisQ: recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of ACM MobiSys*, 2014.

[25] Jeremy Gummeson, Bodhi Priyantha, and Jie Liu. An energy harvesting wearable ring platform for gestureinput on surfaces. In *Proceedings of ACM MobiSys*, 2014.

[26] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-Han Kim. TypingRing: A Wearable Ring Platform for Text Input. In *Proceedings of ACM MobiSys*, 2015.

[27] Fadel Adib, Zach Kabelac, and Dina Katabi. Multi-Person Localization via RF Body Reflections. In *Proceedings of USENIX NSDI*, 2015.

[28] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. Wi-fi gesture recognition on existing devices. http://arxiv.org/ftp/arxiv/papers/1411/1411.5394.pdf.

[29] Heba Abdelnasser, Moustafa Youssef, and Khaled Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *Proceedings of IEEE INFOCOM*, 2015.

[30] IEEE 802.11ac – Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. IEEE P802.11ac/D5.0, 2013.

[31] Ioannis Pefkianakis, Henrik Lundgren, Augustin Soule, Jaideep Chandrashekar, Pascal Le Guyadec, Christophe Diot, Martin May, Karel Van Doorselaer, and Koen Van Oost. Characterizing Home Wireless Performance: The Gateway View. In *Proceedings of IEEE INFOCOM*, 2015.