Robust Indoor Wireless Localization Using Sparse Recovery

Wei Gong and Jiangchuan Liu
School of Computing Science, Simon Fraser University, Canada
{*gongweig, jcliu*}*@sfu.ca*

***Abstract*—With the multi-antenna design of WiFi interfaces, phased array has become a promising mechanism for accurate WiFi localization. State-of-the-art WiFi-based solutions using AoA (Angle-of-Arrival), however, face a number of critical challenges. First, their localization accuracy degrades dramatically when the Signal-to-Noise Ratio (SNR) becomes low. Second, they do not fully utilize coherent processing across all available domains. In this paper, we present ROArray, a RObust Array based system that accurately localizes a target even with low SNRs. In the spatial domain, ROArray can produce sharp AoA spectrums by parameterizing the steering vector based on a sparse grid. Then, to expand into the frequency domain, it jointly estimates the ToAs (Time-of-Arrival) and AoAs of all the paths using multi-subcarrier OFDM measurements. Furthermore, through multi-packet fusion, ROArray is enabled to perform coherent estimation across the spatial, frequency, and time domains. Such coherent processing not only increases the virtual aperture size, which enlarges the number of maximum resolvable paths, but also improves the system robustness to noise. Our implementation using off-the-shelf WiFi cards demonstrates that, with low SNRs, ROArray significantly outperforms state-of-the-art solutions in terms of localization accuracy; when medium or high SNRs are present, it achieves comparable accuracy.**

***Keywords*-WiFi localization; Sparse recovery; AoA;**

I. INTRODUCTION

With the advances in wireless communication and the deep penetration of WiFi networks, WiFi-based localization that aims to deliver GPS-like positioning services for indoor environments has seen rapid growth in the past decade [1], [2]. Early WiFi localization has mainly focused on fingerprinting methods that assume each distinct location has a unique WiFi signature [3], [4], [5]. Despite of meter-level accuracy achieved, they suffer from laborious site survey or demand crowdsourced data that are often not available or of poor quality. More importantly, they typically require dozens of access points (APs) to acquire desirable localization accuracy. Realizing the availability of rich sensor data on advanced smartphones and tablets, sensor-enhanced solutions have been developed to boost the localization accuracy and reduce the demands on APs [6], [7], [8]. They are unfortunately not universal to all-size WiFi clients, in particular, to such thin clients as WiFi-tags [9]. Recently, inspired by the wide deployment of multi-antenna transceivers, phased array with smart signal processing on APs has become a promising mechanism for accurate WiFi localization. In particular, decimeter accuracy can be achieved using Time-of-Arrival (ToA) [2] or Angle-of-Arrival (AoA) [1] techniques.

ToA tracks the signals' time of flight to estimate a client's distance and relative position to APs. The resolution of ToA is fundamentally limited by the narrow bandwidth of WiFi signals. Though higher resolution can possibly be made by the virtual wide band [2], [10], [11], these methods either are not compatible with off-the-shelf devices [11] or rely on channel hopping that inevitably disrupts regular communication [2], [10]. AoA, on the other hand, identifies angles of the multipath signals received at the antenna array of an AP [1], [12], [13]. The typical solution of AoA is done by *multiple signal classification* (MUSIC) [14], which explores the fact that the signal space is orthogonal to the noise space. Such state-of-the-art AoA implementations as SpotFi [1] can achieve a median localization accuracy of 40 cm, and is fully compatible with the current WiFi interfaces. Their practical application and further improvement, however, face several critical challenges.

1) Low SNR barrier. The resolvability of MUSIC inherently degrades when SNR decreases [1]. Particularly, when the noise space is tangled with the signal space, its performance could significantly deteriorate. Although this is a known problem for MUSIC [15], we empirically investigate this aspect in detail as later shown in Section II, which demonstrates that its median accuracy degrades to 15.2° with SNRs lower than 2 dB.

2) Incoherent processing. Usually, multiple OFDM channel measurements contain information from the spatial domain by multiple antennas, from the frequency domain by subcarriers, and from the time domain by a series of consecutive packets. Yet prior systems fail to make the best of them. For example, Ubicarse [8] and ArrayTrack [12] only focus on the spatial domain and time domain. SpotFi [1] coherently performs ToA&AoA estimation but applies clustering, a non-coherent processing, across packets, losing the opportunity to improve SNRs in the time domain.

3) Inefficiency of direct path identification. Most existing methods suffer from being unable to work with a limited number of packets. For instance, SpotFi [1] tends to produce spurious estimates and thus dozens of packets are needed to do clustering; Ubicarse [8] and ArrayTrack [12] need motion on either mobile users or APs to select the stable (or

[1] Assume the size of array and the number of snapshots are fixed.

unchanged) path as the direct path. This inevitably prolongs the localization process. Even worse, *frame aggregation*, which wraps several Ethernet frames into a single frame, has been extensively used to improve the throughput in modern WiFi networks [16]. [2] With frame aggregation, only one CSI (channel state information) measurement is available for multiple frames, and the time cost of localization can thus be significantly amplified.

To address these challenges, this paper presents ROArray, a RObust phased Array based WiFi localization system using off-the-shelf devices. It works with one or a limited number of packets. More importantly, it can reliably locate targets with low SNRs. The design of ROArray is based on a key observation: in an indoor environment, the number of dominant paths is sparse (e.g., 5) [1], [12]. For example, if we divide all possible directions [0°,180°] into an equally spaced sampling grid and the spacing of the grid is 1°, 5 can be safely considered sparse as $5 \ll 180$. Such sparsity is even more obvious when the frequency and spatial domains are considered simultaneously. As such, advanced sparse recovery techniques can be used in this context for AoA and ToA estimation, some of which have been proved robust in noisy cases [17], [18], [19].

Different from MUSIC that focuses on the orthogonality of noise and signal, our concentration is based on the sparsity of signals and coherent processing across the spatial, frequency, and time domains at the same time. First, with multipath, we transform AoA estimation into a sparse recovery problem by parameterizing the space over a sampling grid. By enforcing sparsity on this grid, the resulting AoA spectrum is guaranteed to be sharp and robust. Furthermore, together with the help of OFDM that transmits over a set of subcarriers simultaneously, we jointly estimate the ToAs and AoAs of all the paths and pick up the smallest ToA path as the direct path. Our scheme has several advantages over previous ones [1], [12]. It is insensitive to poor model order (the number of paths) estimates and hence does not suffer from spurious peaks as MUSIC does. It also works with a fairly large operation range, as low as a single packet. Through a coherent combination of information from the spatial, frequency, and time domains, we further improve the spatial resolution by increasing the aperture size, and the robustness to noises by signal decomposition.

We have implemented ROArray on off-the-shelf devices with Intel Ultimate N WiFi Link 5300 cards, and evaluated it in real-world indoor settings. With low SNRs (≤ 2 dB), ROArray achieves a median localization accuracy of 0.91 m, which is remarkably better than that of SpotFi (2.61 m) and ArrayTrack (3.52 m). With medium or high SNRs, ROArray's accuracy is comparable to SpotFi and ArrayTrack; yet it can work well with both a single and multiple measurements, whereas the latter two both require dozens of packets.

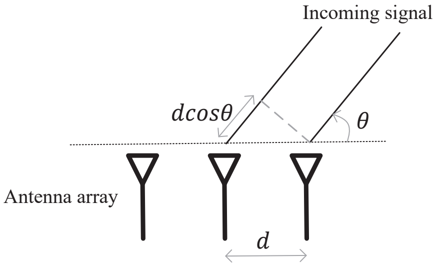


Figure 1. An antenna array consisting of a series of equally spaced antennas. Suppose the AoA of a farfield incoming signal is θ, then the relative phase difference between two adjacent antennas is $-2\pi d\cos\theta/\lambda$, which is due to the difference between two parallel paths, $d\cos\theta$. To avoid ambiguities for $\theta \in [0, 180]$, d needs to be less than or equal to $\lambda/2$, where λ is the wavelength of the incoming signal.

Contributions: To the best of our knowledge, ROArray is the first WiFi localization system that provides robust performance under challenging low-SNR scenarios using off-the-shelf devices.

II. Background and Motivation

To understand the limits of MUSIC, we start from the basics of AoA estimation [12] and then investigate the performance of SpotFi [1], the best-performing AoA implementation, under different SNR scenarios.

A. AoA Estimation Basics

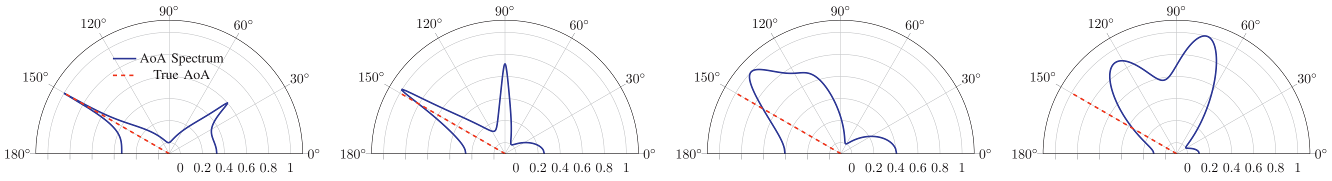
In an indoor environment, a signal usually travels along the direct path and several other reflected paths from a transmitter to a receiver, a.k.a., the multipath effect. Suppose there are K propagation paths. For the k-th path, let θ_k and a_k be the angle and complex attenuation with it respectively. When the signal travels along this path and arrives at the antenna array as shown in Figure 1, the amplitude of attenuation should be almost the same across antennas for the far incoming signal but the phase difference is noticeable among antennas, which depends on θ, λ, and d, where d is the distance between two adjacent antennas. Since d and λ are usually static, the k-th path now can be uniquely decided by a_k and θ_k. Therefore, for an antenna array of size M with an incoming signal at θ_k, those introduced phase shifts relative to the first antenna are given by a vector,

$$s(\theta_k) = [1, \Lambda(\theta_k), ..., \Lambda(\theta_k)^{(M-1)}]^T, \quad (1)$$

where $\Lambda(\theta_k) = \mathrm{e}^{-2\pi d\cos\theta_k/\lambda}$. It shows that an AoA can be viewed as creating a vector of phase shifts on the antenna array, which is why the antenna array is also called a phased array. If we combine those vectors along all the paths, a matrix can be given by

$$\mathbf{S} = [s(\theta_1), s(\theta_2), ..., s(\theta_K)]. \quad (2)$$

[2] 802.11n defines two types of frame aggregation: *MAC Service Data Unit* (MSDU) aggregation and *MAC Protocol Data Unit* (MPDU) aggregation.



(a) High SNR (18 dB)

(b) Medium SNR (7 dB)

(c) Low SNR (2 dB)

(d) Low SNR (<0 dB)

Figure 2. The indoor experimental results of SpotFi under different SNRs. We keep the AoA of direct paths (LoS) fixed at 150° across a range of SNRs. We can see that the performance of SpotFi is very well when SNRs are 18 dB and 7 dB. Nevertheless, when the SNR drops to 2 dB, the estimate is about 12° obviated from the ground truth. The situation is even worse when the SNR is below 0. With the low SNR, the resolvability (the sharpness of beam) is being poor. Note that the power in the y-axis is normalized for all scenarios. For the rest of this paper, we follow this convention unless otherwise stated.

$\mathbf{S}$ is usually called the steering matrix and $s(\theta)$ is the steering vector.

From physics, we know that the received signal vector, $\mathbf{y}$, at the antenna array due to all the paths follows the superposition principle,

$$\mathbf{y} = \mathbf{S}\mathbf{a}, \tag{3}$$

where $\mathbf{a} = [a_1, a_2, ..., a_K]^T$.

To put the above into a typical WiFi system with 3 antennas, the overall attenuations and phase shifts are measured at each subcarrier of each antenna, which are reported as Channel State Information (CSI) values. For example, if the transmitter uses 1 antenna and the receiver with an Intel 5300 WiFi card uses 3 antennas, for each successfully decoded packet, the receiver is able to obtain a CSI matrix

$$\mathbf{C} = \begin{pmatrix} \mathrm{csi}_{1,1} & \mathrm{csi}_{1,2} & \cdots & \mathrm{csi}_{1,30} \\ \mathrm{csi}_{2,1} & \mathrm{csi}_{2,2} & \cdots & \mathrm{csi}_{2,30} \\ \mathrm{csi}_{3,1} & \mathrm{csi}_{3,2} & \cdots & \mathrm{csi}_{3,30} \end{pmatrix}, \tag{4}$$

where $\mathrm{csi}_{i,j}$ denotes the CSI value from the i-th antenna at the j-th subcarrier and is a complex number. Each column of the above matrix can be considered as one realization (snapshot) of $\mathbf{y}$ in Equation 3. Now the key question becomes how to estimate the AoAs of incoming signals, $\mathbf{S}$, with the overall measured matrix $\mathbf{C}$.

B. *Rationale and Caveats*

To answer the above question, state-of-the-art AoA based WiFi localization systems [1], [12], [13] choose MUSIC as their base. The crux of MUSIC is that the signal space is orthogonal to the noise space. Hence, after estimating the noise space via eigen-decomposition, the AoAs can be derived by finding the peaks of an AoA spectrum [3]. Intuitively, the resolvability of MUSIC depends on the SNR [15]. To investigate how MUSIC performs with different SNRs empirically, we have conducted a series of experiments [4]. Using the results of SpotFi [1] as a case in Figure 2, we have two important observations:[5] (1) as SNRs become lower, the beams in AoA spectrums are getting less sharper, which means the resolvability degrades; and (2) the accuracy of AoA estimates becomes much worse when SNRs are low. Such degradation of AoA estimates brought by low SNRs inevitably affects the overall localization accuracy.

To overcome the low SNR barrier with MUSIC (and hence with most today's AoA implementations), it is necessary to find better alternatives, which leads to our design of ROArry that explores the robust performance of sparse recovery [21], [22], [23].

III. ROArray: Design and Optimization

The central question of sparse recovery is how to accurately recover a high-dimensional vector from a small set of measurements with performance guarantees. Consider a system

$$\mathbf{B} = \mathbf{A}\mathbf{x}, \tag{5}$$

where $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{B} \in \mathbb{C}^m$, matrix $\mathbf{A}$ is of size $m \times n$. If the following two conditions are satisfied, I) $\mathbf{x}$ is sparse, $m \ll n$; and II) $\mathbf{A}$ is known, then the above equation can be considered as a sparse recovery problem [23]. Using convex optimization, we can have highly robust results under noisy cases.

As mentioned, in the indoor environment, the number of dominant paths is sparse. Our ROArray explores this opportunity to transform AoA estimation into a sparse recovery problem that satisfies Conditions I and II. Furthermore, we show that after linearization, both ToAs and AoAs can be estimated by enforcing the sparsity constraints. Moreover, ROArray can work well with a limited number of snapshots, even a single packet. Also, it is insensitive to inaccurate initialization, such as $\hat{K}$.

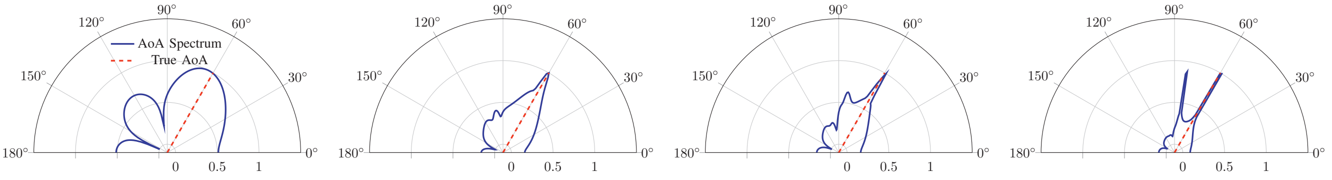
A. *Sparse Recovery for AoA Estimation*

We first revisit Equation 3, $\mathbf{y} = \mathbf{S}\mathbf{a}$. At first glance, all the two conditions are not met. By a proper transformation, we can cast Equation 3 into a sparse recovery problem. The idea

[3] For MUSIC algorithms, please refer to [14], [15], [20].

[4] Details of the experiment settings can be found in section IV.

[5] Although it is difficult to obtain the ground truth AoAs for all paths in practical indoor environments, we use the ground truth of the direct path in Line-of-Sight (LoS) scenarios.



(a) 3 iterations

(b) 6 iterations

(c) 9 iterations

(d) 14 iterations

Figure 3. Progress with iterations using SoC programming to build the AoA spectrum. It can be seen that the iterative procedure improves with more iterations and finally yields a sharp spectrum that gives two AoA estimates, one of which goes well with the ground truth angle.

is to linearize it by expanding the matrix $\mathbf{S}$ via setting up a grid. Specifically, let $\{\widetilde{\theta^1}, \widetilde{\theta^2}, ..., \widetilde{\theta^N}\}$ be an equally spaced grid, spanning over [0°,180°]. In order to meet Condition I, usually N needs to be much greater than M, which is usually 3 in an AP. For example, we can set $N = 181$ if we want the grid spacing to be 1°, or $N = 361$ to obtain an even finer grid. Then, we can construct a new steering matrix consisting of steering vectors that correspond to each element in the grid,

$$\tilde{\mathbf{S}} = [s(\widetilde{\theta^1}), s(\widetilde{\theta^2}), ..., s(\widetilde{\theta^N})]. \tag{6}$$

This way we find that Condition II is also satisfied in the above linearization because the vector $\{\widetilde{\theta^1}, \widetilde{\theta^2}, ..., \widetilde{\theta^N}\}$ is known by the gridding, and each row of $s(\widetilde{\theta^i})$ is predetermined by the manifold of the antenna array as in Equation 1. At the same time, $\mathbf{a}$ should be replaced by $\tilde{\mathbf{a}} = [\tilde{a_1}, \tilde{a_2}, ..., \tilde{a_N}]^T$, in which $\tilde{a_i}$ is nonzero and equal to a_k if some path, e.g., k-th, comes from $\widetilde{\theta^i}$ and is zero otherwise. Note that the major difference between $\mathbf{S}$ and $\tilde{\mathbf{S}}$ is that $\tilde{\mathbf{S}}$ is known and does not depend on any ground truth θ_i.

Now we can cast Equation 3 into a sparse representation,

$$\mathbf{y} = \tilde{\mathbf{S}}\tilde{\mathbf{a}}. \tag{7}$$

To solve the above equation, an important and necessary assumption is the sparsity of $\tilde{\mathbf{a}}$. Fortunately, this assumption is satisfied in indoor WiFi systems as the number of dominant paths is around 5, which is empirically observed in [1], [12], [13]. The ideal measurement of the sparsity of a vector is the ℓ_0 norm, $\|\tilde{\mathbf{a}}\|_0$. However, solving $\min \|\tilde{\mathbf{a}}\|_0$ such that $\mathbf{y} = \tilde{\mathbf{S}}\tilde{\mathbf{a}}$ is quite hard and almost intractable even when $\tilde{\mathbf{a}}$ is of moderate size. Hence, we employ one of the well-known approximations for this problem, using ℓ_1 norm to approximate ℓ_0 norm. The rationale behind this is that it has been proved that if $\tilde{\mathbf{a}}$ is sparse enough, this approximation actually can lead to exact solutions [21].

So far we have not discussed noises, which are inevitable in practice. Considering the additive Gaussian noise, the model in Equation 7 becomes

$$\mathbf{y} = \tilde{\mathbf{S}}\tilde{\mathbf{a}} + \mathbf{n}. \tag{8}$$

In noisy cases, our objective is to solve the following optimization problem,

$$\min \|\tilde{\mathbf{a}}\|_1, \tag{9}$$

$$\text{s.t.} \|\mathbf{y} - \tilde{\mathbf{S}}\tilde{\mathbf{a}}\|_2^2 \leq \gamma, \tag{10}$$

where γ is a parameter to specify the level of noise the sytem can tolerate. We can reformulate the above equations using the method of Lagrange multipliers as,

$$\min \|\mathbf{y} - \tilde{\mathbf{S}}\tilde{\mathbf{a}}\|_2^2 + \kappa\|\tilde{\mathbf{a}}\|_1, \tag{11}$$

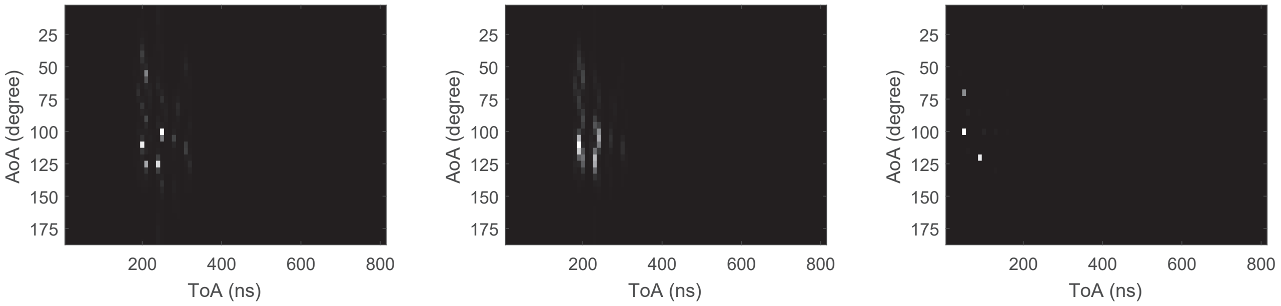
where κ is a parameter used to enforce the level of sparsity. It is easy to verify that the above objective function is convex, which means we can make use of second-order cone (SoC) programming [6] to efficiently solve the problem [22]. One salient feature of using SoC programming is that the number of iterations in the worst case is bounded [22]. Once $\tilde{\mathbf{a}}$ is found, the AoA estimates are the peaks in $\tilde{\mathbf{a}}$. An illustrative example is given in Figure 3. Note that ℓ_1 based algorithms have global convergence regardless of the initialization [21], i.e., insensitive to initialization.

B. Direct Path Identification

After we obtain AoA estimates from the previous section, to localize the target, we need to distinguish the direct path from other reflected paths. State-of-the-art AoA based systems are all based on dozens of measurements or motion to pick up the stable (unchanged) path with the smallest variation as the direct path. In contrast, we intend to jointly estimate ToAs and AoAs for all the paths and pick up the direct path that is with the smallest ToA. Note that LTEye [24] shares the same direct path identification criteria with ours, but it requires multi-packet measurements through a motorized array whereas our scheme does not require motion of targets or APs.

From physics, we know that each independent propagation path comes with a distinct ToA and AoA. For a narrow band signal, ToAs are usually omitted as they introduce no noticeable phase shift. However, the OFDM WiFi consists of a number of narrow bands (subcarriers), where phase shifts brought by ToAs are not negligible. Particularly, we

[6] This is because our data is complex. For real data, it can readily be solved using quadratic programming.



(a) ToA&AoA spectrum from packet A (b) ToA&AoA spectrum from packet B (c) ToA&AoA spectrum from 30 packets

Figure 4. The estimated ToA&AoA spectrums from two time samples are presented in (a), (b). Even with the same ground truth (both the transmit and receiver are static), both spectrums are associated with different packet detection delays. After delay estimation and multi-packet fusion, the result, (c), becomes sharper (more accurate).

observe that AoAs introduce no much difference across subcarriers, but ToAs contribute to measurable phase shifts across subcarriers. For example, for the k-th path, the phase shift introduced across two subcarriers (f_i, f_j) spaced by 20 MHz is $-2\pi d \cos\theta_k (f_i - f_j)/c$, where c is the speed of light, so if $d = \lambda/2$ and $\lambda = 5.2$ cm for 5 GHz band, the maxima of this phase shift ($\theta_k = 0$) only amounts to 0.0054 radians, which is too small to measure. By contrast, even if the ToA of the k-th, τ_k, is only 5 ns, the phase shift introduced by the same two subcarriers of an antenna along this path, is $-2\pi(f_i - f_j)\tau_k = 0.628$ radians, which is much greater than 0.0054 radians. Therefore, this motivates us to jointly estimate ToAs and AoAs for paths in OFDM WiFi systems as either ToA or AoA alone is not enough to account for overall phase shifts. Hence, by including all the subcarriers in WiFi, we can remodel the narrow-band steering vector $s(\theta)$ to a new joint ToA and AoA steering vector $s(\theta, \tau)$. Specifically, we know that for a path with τ_k, the phase shift introduced between two adjacent subcarriers is

$$\Gamma(\tau_k) = \mathrm{e}^{-2\pi f_\delta \tau_k}, \tag{12}$$

where f_δ is the spacing of two adjacent subcarriers [7]. Then we stack the steering vectors across subcarriers into a new steering vector that is represented by different phase shifts caused by ToAs and AoAs,

$$s(\theta, \tau) = [\underbrace{1, \Lambda_\theta, ..., \Lambda_\theta^{M-1}}_{\text{subcarrier 1}}, ..., \underbrace{\Gamma_\tau^{L-1}, \Lambda_\theta \Gamma_\tau^{L-1}, ..., \Lambda_\theta^{M-1}\Gamma_\tau^{L-1}}_{\text{subcarrier L}}]^T, \tag{13}$$

where L is the number of measured subcarriers. Then similar to the previous section, we can further linearize this new steering vector by setting up two grids: $\{\tilde{\tau^1}, \tilde{\tau^2}, ..., \tau^{\tilde{N}_\tau}\}$ for ToA, and $\{\tilde{\theta^1}, \tilde{\theta^2}, ..., \theta^{\tilde{N}_\theta}\}$ for AoA. The range of this ToA grid is $[0, \tau_{max}]$, where $\tau_{max} = 1/f_\delta$. For example, if Intel 5300 cards work with a 40 MHz band, then $1/f_\delta = 1.25$ MHz and thus $\tau_{max} = 800$ ns.

Specifically, considering CSI values from Intel 5300 cards, to jointly estimate ToAs and AoAs, we stack all the subcarrier measurements and linearized steering matrices together as follows

$$\mathbf{y}_{\theta\tau} = \mathbf{S}_{\theta\tau}\mathbf{a}_{\theta\tau} + \mathbf{n}, \tag{14}$$

$$\mathbf{y}_{\theta\tau} = [\underbrace{\mathrm{csi}_{1,1}, \mathrm{csi}_{2,1}, \mathrm{csi}_{3,1}}_{\text{subcarrier 1}}, ..., \underbrace{\mathrm{csi}_{1,30}, \mathrm{csi}_{2,30}, \mathrm{csi}_{3,30}}_{\text{subcarrier 30}}]^T, \tag{15}$$

$$\mathbf{S}_{\theta\tau} = [\underbrace{s(\tilde{\theta^1}, \tilde{\tau^1}), ..., s(\theta^{\tilde{N}_\theta}, \tilde{\tau^1})}_{\text{size of } 90\times N_\theta}, ..., \underbrace{s(\tilde{\theta^1}, \tau^{\tilde{N}_\tau}), ..., s(\theta^{\tilde{N}_\theta}, \tau^{\tilde{N}_\tau})}_{\text{size of } 90\times N_\theta}], \tag{16}$$

$$\mathbf{a}_{\theta\tau} = [a_1, ..., a_{N_\theta N_\tau}]^T, \tag{17}$$

where $\mathbf{y}_{\theta\tau}$ and $s(\theta, \tau)$ are of size 90×1, $\mathbf{S}_{\theta\tau}$ is of size $90 \times N_\theta N_\tau$, and $\mathbf{a}_{\theta\tau}$ is of size $N_\theta N_\tau \times 1$. Since all the K paths are sparse both in AoA and ToA domains, solving the above equation equals to

$$\min \|\mathbf{y}_{\theta\tau} - \mathbf{S}_{\theta\tau}\mathbf{a}_{\theta\tau}\|_2^2 + \kappa\|\mathbf{a}_{\theta\tau}\|_1. \tag{18}$$

This way, the estimated spectrum can result in a desirable sharpness for both ToAs and AoAs, as shown in Figure 4a.

Once we obtain the AoAs and ToAs of all paths, we just pick up the smallest ToA path as the direct path. This idea has been evaluated and deemed as a suboptimal approach in SpotFi [1]. Actually, we have found it does not suit SpotFi as MUSIC tends to produce spurious peaks due to inaccurate $\hat{K}$. Nevertheless, it just fits our ROArray, because our method is not sensitive to inaccurate $\hat{K}$. Note that there is another important benefit brought by stacking all the subcarriers into a single vector: the increasd aperture size of the antenna array, which makes the number of resolvable paths more than M. Before stacking, the aperture size of an antenna array is always limited by the number of antennas

[7] In typical WiFi standards like 802.11n/ac, $f_\delta = 312.5$ KHz. However, in practice, f_δ depends on measured CSI values in different systems. For example, for Intel 5300 cards that report CSI values every 4 subcarriers on a 40 MHz band, $f_\delta = 1.25$ MHz.

(usually 3), where the number of resolvable paths is less than M.

C. Complexity

In terms of time complexity, to solve Equation 18, it requires $\mathcal{O}((N_\theta N_\tau)^3)$ while being almost independent of the number of antennas, M, and the number of subcarriers, N_{sub}. This time complexity is higher than that of SpotFi, $\mathcal{O}((MN_{sub})^3)$. But thanks to the interior point method that has low iteration times [23], a general implementation can be quite efficient. For instance, our Matlab implementation with an Intel i7 CPU at 3.4 GHz takes about 10 s to generate a ToA&AoA spectrum, when $N_\theta = 90$, $N_\tau = 50$. The optimization of computation time is one of our future work. Actually, we think of this higher computation cost as the tradeoff between accuracy and computation time because the better performance under low SNRs does not come for free. Despite the higher time cost, some advantages are worth noting. First, it is suitable for applications that concern accuracy more than computation time/power, especially when low SNRs are present. Second, ROArray is more general as it is not affected too much by the array geometry, such as rigid antenna placement, unevenly distributed bands, and the number of antennas, making the adaptation to other WiFi standards easier, like 802.11 ac/ad.

D. Implementation considerations

There are several considerations worth noting here.

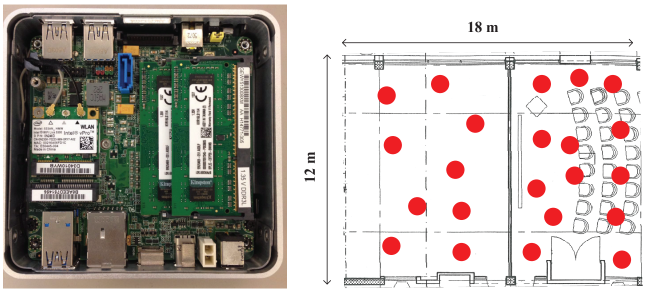
Multi-Packet fusion. While ROArray can work with a single packet, it can also leverage multi-packet measurements to further improve the accuracy for slowly moving and static objects. Different from prior methods that either treat each packet independently and then use clustering to filter outliers [1], we adapt the method in [25] and use the Singular Value Decomposition (SVD) to simultaneously reduce the problem size efficiently and maintain the high performance.

Multi-AP localization. For localization, ROArray attempts to localize the target by combing direct-path AoAs from several APs. Let $\{\varphi_1, \varphi_2, ..., \varphi_l\}$ be the estimated AoAs from l APs and $\{R_1, R_2, ..., R_l\}$ be the corresponding RSSIs. ROArray intends to localize the target by minimizing the deviation between RSSI-weighted AoAs,

$$\min \sum_{i=1}^{l} R_i(\hat{\varphi}_i - \varphi_i)^2. \tag{19}$$

Hence, we search the candidate area by forming a 10 cm by 10 cm grid and pick up the location that achieves the minimal of the above equation.

Phase calibration. Every time when the working channel changes, a random phase offset will be introduced in measured CSI values. We adapt autocalibration algorithms used in [13] to correct phases in ROArray. The major difference is that we use ROArray's estimated AoA spectrums instead of MUSIC in [13]. The involved cost of this calibration would be negligible as it is only invoked when the AP administrator sets or changes the channel, and there is no channel hopping involved in regular WiFi communication.



(a) An Intel NUC Unit with an Intel 5300 WiFi card

(b) Experiment testbed

Figure 5. Experiment device and deployment. Our experiments involve several APs and a mobile client that is an Intel NUC. The testbed covers 18 m × 12 m indoor area. Red dots represent test locations.

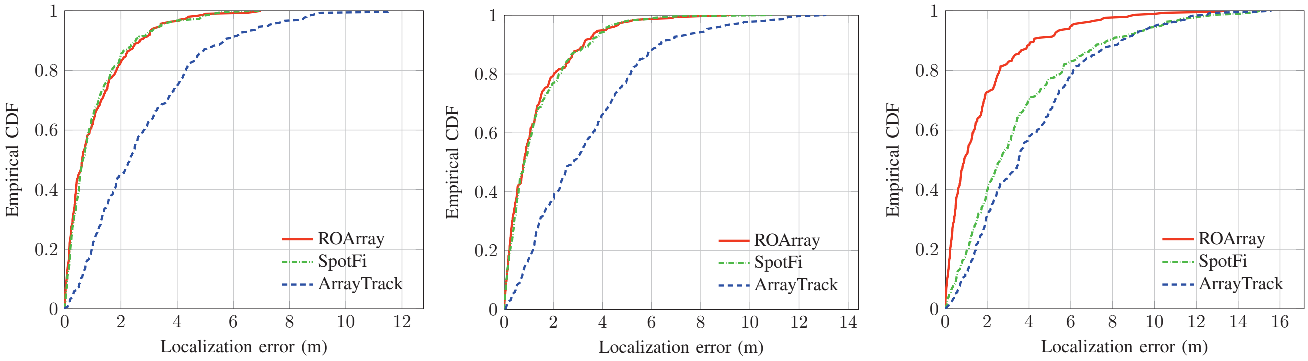
IV. Experiments

A. Implementation

To verify the design of ROArray, we implement it using off-the-shelf Intel 5300 WiFi cards. Linux CSI Tools [26] are employed to obtain CSI measurements. Due to firmware limitations that produce phase ambiguity on 2.4 GHz band [1], [13], all tests are done in 5 GHz band. We randomly choose a non-busy 40 MHz channel in our testbed and fix it during all tests. Note that Intel 5300 cards give CSI values only for 30 out of 116 subcarriers. We employ 6 desktops working as APs and one Intel NUC unit as a mobile client. All the APs and the client are equipped with Intel 5300 cards. Each AP is with 3 antennas that are equally spaced at half wavelength, 2.6 cm. All the APs work in the monitor mode while the client uses packet injection to send out data. We set MCS index at 1 for all packets, which means 1 spatial stream, QPSK modulation, and 1/2 coding rate.

After the client sends out a packet, APs transmit measured CSI values to a central server for further processing. The central server synchronizes measurements by matching sequence numbers in the payload. Then it runs MATLAB-implemented algorithms to obtain estimated locations. We use cvx solvers [27] to deal with the sparse recovery problem and note that the code speed could be further optimized. We have tested our prototype in a classroom testbed and part of the tested locations are marked in Figure 5. In total, we tested 300 different locations.

We compare ROArray with state-of-the-art AoA based WiFi systems, SpotFi [1] and ArrayTrack [12]. Phaser [13] is not included because it requires additional hardware, such as antenna rerouting, and same as WiDeo [28], which requires software defined radios to implicitly eliminate packet detection delay (and the fractional SFO), which is not compatible with Intel 5300 cards. Ubicarse [8] and CUPID [6] are not



(a) High SNRs, $\geq$15 dB

(b) Medium SNRs, (2,15) dB

(c) Low SNRs, $\leq$2 dB

Figure 6. Comparison of ROArray's localization accuracy with SpotFi and ArrayTrack under high, medium, and low SNR scenarios.

compared either, as they need inertial sensors and mobility of clients, whereas ROArray, SpotFi, and ArrayTrack do not have such assumptions and are readily implementable to all kinds of devices that have WiFi. Note that although original Arraytrack implementation requires 6-8 antennas and software-defined radios, we implement its algorithms using the aforementioned hardware settings (3-antenna), ensuring a fair competition.

B. Localization Accuracy Comparison

Dynamic SNRs actually are quite common in indoor multipath-rich environments, since multipath could result in constructive or destructive interferences. Also, SNRs are affected by blocking, distance, and the transmit power of APs. Here we do not distinguish the causes of dynamic SNRs and just classify SNRs into three categories, high SNRs $[15, \infty)$, medium SNRs $(2, 15)$, and low SNRs $(-\infty, 2]$. All three methods share the same data and each uses 15 packets. We fix the number of APs at 6 for this comparison and report results in Figure 6.

We observe from Figure 6a that with high SNRs, ROArray accomplishes comparable results with SpotFi and significantly outperforms ArrayTrack. Particularly, ROArray achieves 0.63 m median localization error while SpotFi and ArrayTrack's median accuracy is 0.64 m and 2.3 m, respectively. And the 90-th percentile errors are 2.66 m, 2.51 m, and 5.66 m for ROArray, SpotFi, and ArrayTrack respectively. The reason for ArrayTrack's relatively poor performance is that its aperture size is very limited, in contrast, ROArray and SpotFi increase the aperture size by coherently combining CSI values across subcarriers. Similar trends can be observed with medium SNRs in Figure 6b. As expected, when SNRs decrease, the performances of all the three systems deteriorate accordingly. However, when SNRs drops to a low level as shown in Figure 6c, the median accuracies of SpotFi and ArrayTrack degrade to 2.61 m and 3.52 m respectively whereas ROArray achieves 0.91 m. Such performance gain of ROArray over other systems largely comes from the robustness of sparse recovery techniques.

C. Direct Path Accuracy Comparison

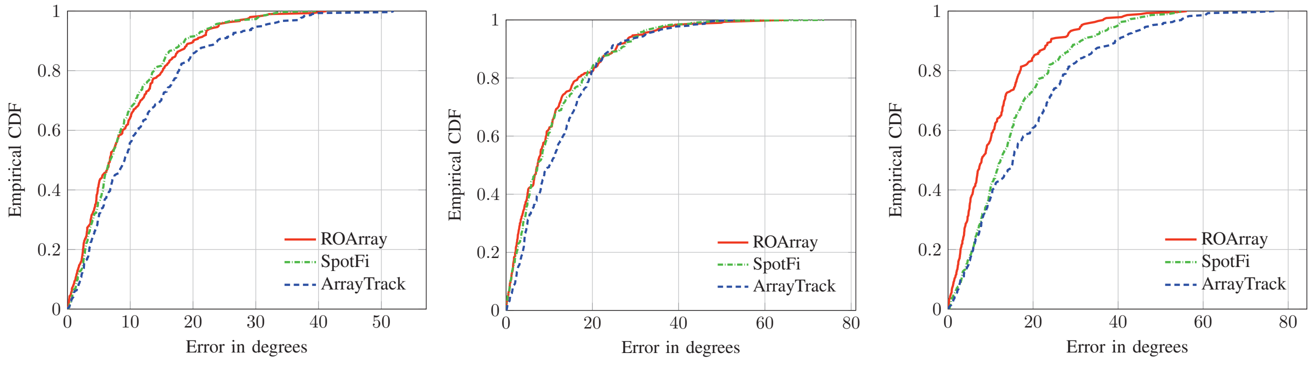
To further examine the resolvability of three different systems, we investigate AoA estimate errors. Since we do not have the ground truth AoAs for all the paths, we measure the accuracy of AoA estimation algorithms by comparing the difference between the ground truth direct-path AoA and the closest peaks in the spectrum.

For this test, we still conduct the evaluation with three different SNR situations. Figure 7a plots the CDFs of AoA estimation errors for all APs with high SNRs, where ROArray achieves almost the same median AoA accuracy as SpotFi, which is 2.48 degrees better than ArrayTrack. When SNRs go into the medium level as shown in Figure 7b, the degradation of all the three systems is quite limited as the median AoA accuracies worsen from 6.7 to 7.32 degrees for ROArray, from 6.62 to 7.40 degrees for SpotFi, and from 9.10 to 10.0 degrees for ArrayTrack. This phenomenon again shows good performance of MUSIC with high and medium SNRs and confirms that sparse recovery algorithms are robust. Note that in low-SNR situations, the median accuracy of ROArray only drops to 7.9 degrees whereas those of SpotFi and ArrayTrack degrade to 12.3 degrees and 15.2 degrees respectively. Even the aperture size of SpotFi is the same as ROArray's, the inherent drawback of MUSIC that relies on the separation of the signal and noise spaces makes SpotFi less robust compared to ROArray. Another factor that contributes to the inaccuracy for SpotFi is its sensitivity to inaccurate $\hat{K}$ [8].

D. Varying Number of APs

Next, we investigate how AP density impacts the accuracy for ROArray. By varying the number of APs that can hear

[8] In fact, SpotFi fixes $K = 5$ [1], which intuitively cannot adapt to various ground truth K.

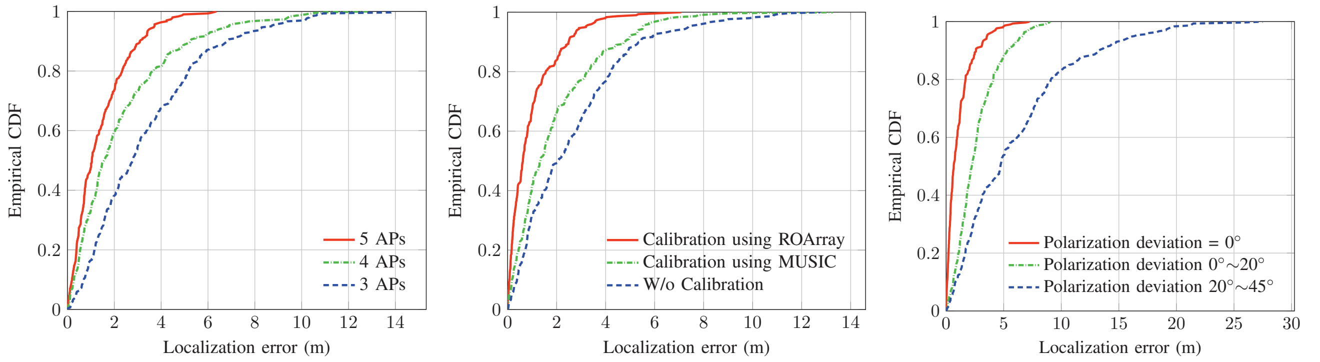


(a) High SNRs, ≥15 dB

(b) Medium SNRs, (2,15) dB

(c) Low SNRs, ≤2 dB

Figure 7. Comparison of ROArray's AoA estimation errors with SpotFi and ArrayTrack under high, medium, and low SNR scenarios.



(a) Localization errors with different # of APs

(b) Localization errors with different phase calibration schemes

(c) Localization errors with different polarization deviation angles

Figure 8. (a) CDFs of localization errors for ROArray with varying number of APs, showing how AP density affects ROArray's performance; (b) CDFs of localization errors for ROArray with different calibration schemes, demonstrating the advantage of the calibration scheme using the AoA spectrum of ROArray; (c) CDFs of localization errors for ROArray with different deviation angles of antenna polarization, indicating the importance of polarization.

the client from 3 to 5, we present results in Figure 8a. We observe that accuracy improves with the increasing density of APs, similar to other AoA based systems [1], [12]. Because with more APs, our RSSI-weighted localization scheme tends to give greater weights to high-quality direct paths, largely reducing the negative impact brought by noisy estimates. We see that the median accuracies of ROArray are 1.04 m, 1.56 m, and 2.79 m with 5, 4, and 3 APs respectively. Note that, with only 3 APs, the performance of ROArray almost catches up with that of ArrayTrack.

E. Impact of Phase Calibration

Moreover, we evaluate the impact of different phase calibration schemes. In this experiment, we perform localization in three different ways, phase calibration using ROArray's AoA spectrum, phase calibration using MUSIC based AoA spectrum [13], and without any phase calibration. The results are plotted in Figure 8b. Unsurprisingly, the way that is without any phase calibration performs the worst and only has a median accuracy of 2.0 m. ROArray's calibration achieves an improvement of 0.71 m over the scheme in Phaser [13] in terms of median localization accuracy. This huge performance gain stems from the sharper AoA spectrum of ROArray over that of MUSIC. In fact, such sharpness should be further attributed to that we use sparse recovery techniques for the AoA estimation problem.

F. Impact of Antenna Polarization

For mobile users, the orientation of antenna keeps changing. To measure the effects of deviation angles of antenna polarization, we use horizontally polarized antennas on APs and randomly deviate the elevation angles of the mobile client between (0°, 20°] and (20°, 45°]. The results in Figure 8c show that the accuracy of ROArray is heavily affected by the deviation angle of polarization. The performance of ROArray worsens with increasing deviation angles. The median localization errors degrade to 2.21 m and 4.71 m for 0°∼20° and 20°∼45° deviation respectively. In fact, this is not surprising as the elevation-angle deviation inevitably leads to very poor wireless reception since the manifold of the antenna array in our implementation is 1-dimension. One possible solution is to employ the 2-dimension antenna array with both vertical and horizontal polarizations, which can adapt to more antenna orientations in 3-D space.

V. Related Work

There has been extensive research in the literature on WiFi based localization systems, so we only summarize closely related ones here. For more complete surveys, please refer to [29], [30]. Basically, there are two types of WiFi localization solutions, signal processing based and RSSI based.

Signal processing based: With the development of multi-antenna design, WiFi localization using signal process techniques has received increasing attention [1], [2], [10], [11], [12], [13]. Technically, signal processing based methods are able to estimate two important metrics, ToA and AoA. Due to the intrinsically limited bandwidth of WiFi signals, the accuracy of ToA based systems is quite limited. Recently, several channel hopping mechanisms are proposed to combine a number of narrow bands into a virtual wide band [2], [10], [11], breaking the barrier of meter-level accuracy. There are two disadvantages associated with those methods: communication disruption due to channel hopping and slow adaptation to moving mobile clients due to multi-channel measurements. On the other hand, AoA based approaches can maintain communication unaffected and realize high accuracy localization [1], [12], [13]. The most recent advance, SpotFi can even achieve 40 cm median accuracy [1]. Another closely related method is WiDeo [28], which also uses sparse recovery to retrieve ToA and AoA information. There are several key differences between ROArray and WiDeo. First, WiDeo achieves better accuracy than ROArray does but it requires software defined radios (SDR) and 4 antennas for each AP. One of the main reasons for the SDR requirement is that it can make use of pilot subcarriers to calibrate uncertain time delays, which means its ToA estimates can be included directly into the location estimation. In contrast, our ToA estimates using off-the-shelf APs contain the residual dynamic delay in each packet [1], for which no solutions and calibration methods are known yet. Second, unlike the very brief statement of actual sparse recovery algorithms in WiDeo, we provide the core and analysis of joint ToA/AoA estimation in detail. Third, the computation time of WiDeo is more than ROArray due to the continuous basis for WiDeo whereas ROArray is based on the discrete basis. Last but not least, we include multi-packet fusion to coherently improve the localization accuracy while the main goal of WiDeo is to trace the motion by identifying static reflections. In summary, prior systems that work with off-the-shelf devices would suffer from poor and unstable performance under low-SNR scenarios, such as far way from APs, serious NLoS, and interference. ROArray falls into the AoA category. While it maintains comparable performance that state-of-the-art systems achieve at high SNRs, it shows robustness with low SNRs. It can also localize a target with one and more packets, making it more applicable in mobile scenarios.

RSSI based: Due to the high availability of RSSI values, RSSI based systems have been studied for many years. Those methods either measure the range via a propagation model [3] or collect fingerprints from a series of APs to locate a client [5]. Nevertheless, the biggest problem for them is not-so-impressed accuracy. The most common way to boost accuracy is to increase the number of APs, usually several dozens. Another drawback is the cost of site survey [30]. When environments or APs change, the system usually needs another overhaul calibration, which is always time-consuming. Even a variety of crowdsourcing schemes can help [4], [5], the quality of crowdsourced data needs extra care.

Sparse recovery based AoA estimation is extensively studied in signal processing and information theory areas [25], [31], [32], which also inspires this work. Those works mainly focus on the theory or numerical validation aspects, not for practical systems, including WiFi. They usually do not take into account practical challenges, such as time/frequency synchronization and phase calibration. Moreover, they do not consider the multi-carrier feature of signals, which is the core of today's WiFi. In contrast, ROArray builds on those well-studied theory results and further develops a working WiFi localization system using off-the-shelf devices.

VI. Conclusion

We have presented a robust WiFi localization system, ROArray, that addressed the poor performance with low SNRs, which was difficult for state-of-the-art approaches. The insight of ROArray was to cast AoA estimation into a sparse recovery problem, which has been able to yield sharp and sparse AoA spectrum. Through jointly estimating the ToAs and AoAs of all the paths across the time domain, we have achieved direct path identification and the increase of resolvability at the same time. We believe ROArray can benefit a range of indoor applications that require high robustness in challenging low-SNR scenarios, such as localization solutions for enterprise and military.

Acknowledgment

We would like to thank the anonymous reviewers for valuable and insightful comments. This work was supported in part by NSFC under Grant No. 61472268, in part by the Canada Technology Demonstration Program, in part by a Canada NSERC Discovery Grant, and in part by the NSERC E.W.R. Steacie Memorial Fellowship.

References

[1] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter Level Localization Using WiFi," in *Proc. of ACM SIGCOMM*, 2015.

[2] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-Level Localization with a Single WiFi Access Point,” in *Proc. of USENIX NSDI*, 2016.

[3] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor Localization Without the Pain,” in *Proc. of ACM MobiSys*, 2010.

[4] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, “Centaur: Locating Devices in an Office Environment,” in *Proc. of ACM MobiCom*, 2012.

[5] Z. Yang, C. Wu, and Y. Liu, “Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention,” in *Proc. of ACM MobiCom*, 2012.

[6] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, “Avoiding Multipath to Revive Inbuilding WiFi Localization,” in *Proc. of ACM MobiSys*, 2013.

[7] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, “SAIL: Single Access Point-Based Indoor Localization,” in *Proc. of ACM MobiSys*, 2014.

[8] S. Kumar, S. Gil, D. Katabi, and D. Rus, “Accurate Indoor Localization With Zero Start-up Cost,” in *Proc. of ACM MobiCom*, 2014.

[9] “Wi-Fi Tags,” http://www.ekahau.com/real-time-location-system/technology/wi-fi-tags.

[10] Y. Xie, Z. Li, and M. Li, “Precise Power Delay Profiling with Commodity WiFi,” in *Proc. of ACM MobiCom*, 2015.

[11] J. Xiong, K. Sundaresan, and K. Jamieson, “ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization,” in *Proc. of ACM MobiCom*, 2015.

[12] J. Xiong and K. Jamieson, “ArrayTrack: A Fine-Grained Indoor Location System,” in *Proc. of USENIX NSDI*, 2013.

[13] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, “Phaser: Enabling Phased Array Signal Processing on Commodity WiFi Access Points,” in *Proc. of ACM MobiCom*, 2014.

[14] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[15] P. Stoica and N. Arye, “MUSIC, Maximum Likelihood, and Cramer-Rao Bound,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 5, pp. 720–741, 1989.

[16] S. Byeon, K. Yoon, O. Lee, S. Choi, W. Cho, and S. Oh, “MoFA: Mobility-aware Frame Aggregation in Wi-Fi,” in *Proc. of ACM CoNext*, 2014.

[17] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[19] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, “Sensitivity to basis mismatch in compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2182–2195, 2011.

[20] P. Stoica and R. L. Moses, *Introduction to spectral analysis*. Prentice hall Upper Saddle River, 1997, vol. 1.

[21] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[22] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, “Applications of second-order cone programming,” *Linear algebra and its applications*, vol. 284, no. 1, pp. 193–228, 1998.

[23] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[24] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, “LTE Radio Analytics Made Easy and Accessible,” in *Proc. of ACM SIGCOMM*, 2014.

[25] D. Malioutov, M. Çetin, and A. S. Willsky, “A Sparse Signal Reconstruction Perspective for Source Localization With Sensor Arrays,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool release: gathering 802.11 n traces with channel state information,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[27] “CVX solvers,” http://cvxr.com/cvx/doc/solver.html.

[28] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, “Wideo: Fine-grained device-free motion tracing using rf backscatter,” in *Proc. of USENIX NSDI*, 2015.

[29] I. Guvenc and C.-C. Chong, “A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[30] Z. Yang, Z. Zhou, and Y. Liu, “From rssi to csi: Indoor localization via channel response,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 25, 2013.

[31] Z. Yang, L. Xie, and C. Zhang, “Off-Grid Direction of Arrival Estimation Using Sparse Bayesian Inference,” *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 38–43, 2013.

[32] M. M. Hyder and K. Mahata, “Direction-of-Arrival Estimation using a Mixed l2,0 Norm Approximation,” *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4646–4655, 2010.