RoArray: Towards More Robust Indoor Localization Using Sparse Recovery with Commodity WiFi

Wei Gong, *Member, IEEE* and Jiangchuan Liu, *Fellow, IEEE*

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 Angle-of-Arrival (AoA), however, face a number of critical challenges. First, their localization accuracy degrades dramatically due to low Signal-to-Noise Ratio (SNR) and incoherent processing. Second, they tend to produce outliers when the available number of packets is low. Moreover, the prior phase calibration schemes are not multipath robust and accurate enough. All of the above degrade the robustness of localization systems. In this paper, we present ROArray, a RObust Array based system that accurately localizes a target even with low SNRs. The key insight of ROArray is to use sparse recovery and coherent processing across all available domains, including time, frequency, and spatial domains. Specifically, 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 Time-of-Arrival (ToAs) and AoAs of all the paths using multi-subcarrier OFDM measurements. Furthermore, through a novel multi-packet fusion scheme, ROArray is enabled to perform coherent estimation over multiple packets. 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. In addition, ROArray includes an online phase calibration technique that can eliminate random phase offsets while keeping communication uninterrupted. 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.

Index Terms—Backscatter communication, frequency selection, RFID, tags

1 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]. In spite 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

- *W. Gong is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230000, China. E-mail: weigong@ustc.edu.cn.*
- *J. Liu is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: jcliu@sfu.ca.*

Manuscript received 26 Apr. 2017; revised 28 Mar. 2018; accepted 12 July 2018. Date of publication 25 July 2018; date of current version 2 May 2019. (Corresponding author: Jiangchuan Liu.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2860018

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

a known problem for MUSIC [15], we empirically investigate this aspect in detail as later shown in Section 2, 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. Nevertheless, 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 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 single channel state information (CSI) measurement is available for multiple frames, and the time cost of localization can thus be significantly amplified.
4) *Offline and multipath-susceptible Phase Calibration.* Prior phase calibration schemes are either designed for offline correction or are not robust to multipath effects. For example, both ArrayTrack [12] and Argos [17] can obtain accurate phase offset estimation through additional offline measurements, yet it is hard to apply them to commercial off-the-shelf (COTS) devices as they are originally designed for software defined radios. Phaser [13] can work with COTS APs, but its performance could severely degrade in multipath-rich environments.

To address these challenges, this paper presents ROArray, a ROobust 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 [18], [19], [20].

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

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. Moreover, to overcome the uncertain dynamic delay and lack of sparsity across packets, a novel coherent ToA and AoA estimation based on multi-packet fusion is introduced. To eliminate random phase offsets each time the working channel changes or powers on/off, we introduce an online phase calibration method, which leverages the orthogonality between steering vectors and the noise space. ROArray 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 5,300 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.

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. Our system applies to static or slow-moving targets of which localization accuracy is more concerned than real-time tracking needs with low SNRs.

2 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.

2.1 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 kth 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 Fig. 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

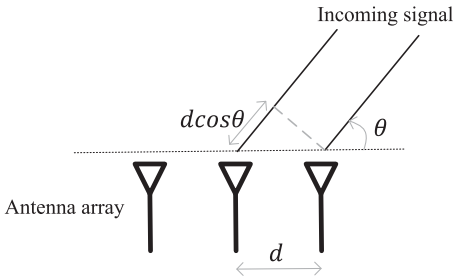


Fig. 1. An antenna array consisting of a series of equally spaced antennas. Suppose the AoA of a far-field incoming signal is θ, then the relative phase difference between two adjacent antennas is $-2\pi d\cos\theta/\lambda$, which is due to the difference between two parallel paths, $d\cos\theta$. To 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.

the distance between two adjacent antennas. Since d and λ are usually static, the kth 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,

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

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

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

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

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 values. For example, if the transmitter uses 1 antenna and the receiver with an Intel 5,300 WiFi card uses 3 antennas, for each successfully decoded packet, the receiver is able to obtain a CSI matrix

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

where $\mathrm{csi}_{i,j}$ denotes the CSI value from the ith antenna at the jth 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 $\mathbf{C}$.

2.2 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 Fig. 2, we have two important observations:[5] (1) as SNRs become lower, the beams in AoA spectrums are getting less sharp, 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 [22], [23], [24].

3 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 [24]. 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 $\hat{K}$.

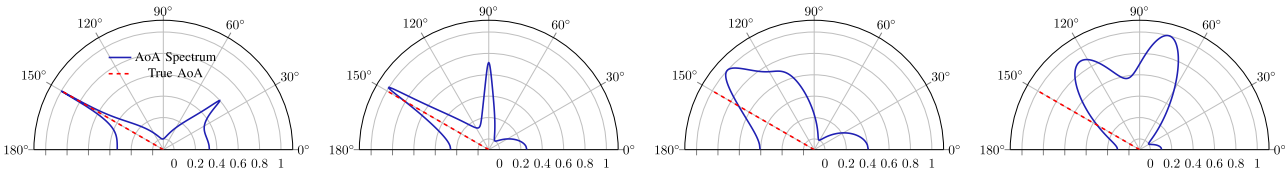
3.1 Sparse Recovery for AoA Estimation

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 is to linearize it by expanding the matrix $\mathbf{S}$ via setting up a grid. Specifically, let $\{\tilde{\theta^1}, \tilde{\theta^2}, \ldots, \tilde{\theta^N}\}$ 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,

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

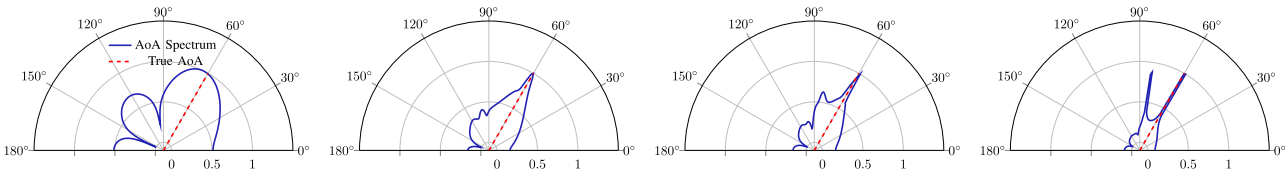
This way we find that Condition II is also satisfied in the above linearization because the vector $\{\tilde{\theta^1}, \tilde{\theta^2}, \ldots, \tilde{\theta^N}\}$ is

3. For MUSIC algorithms, please refer to [14], [15], [21].
4. Details of the experiment settings can be found in Section 4.
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) High SNR (18 dB) (b) Medium SNR (7 dB) (c) Low SNR (2 dB) (d) Low SNR (<0 dB)

Fig. 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.



(a) 3 iterations (b) 6 iterations (c) 9 iterations (d) 14 iterations

Fig. 3. Progress with iterations using SoC programming to build the AoA spectrum. 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.

known by the gridding, and each row of $s(\tilde{\theta^i})$ is pre-determined by the manifold of the antenna array as in Equation (1). At the same time, $\mathbf{a}$ should be replaced by $\tilde{\mathbf{a}} = [\tilde{a_1}, \tilde{a_2}, \ldots, \tilde{a_N}]^T$, in which $\tilde{a_i}$ is nonzero and equal to a_k if some path, e.g., kth, comes from $\tilde{\theta^i}$ 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 if $\tilde{\mathbf{a}}$ is sparse enough, this approximation actually can lead to exact solutions [22].

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 system can tolerate. We can reformulate the above equations using the method of Lagrange multipliers as,

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

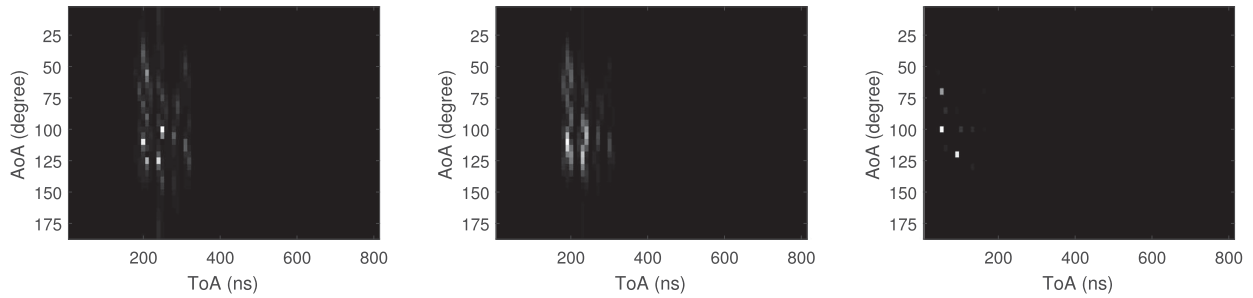
where κ is a parameter used to enforce the level of sparsity. It is easy to verify that the above objective function is convex, which means we can make use of second-order cone (SoC) programming[6] to efficiently solve the problem [23]. One salient feature of using SoC programming is that the number of iterations in the worst case is bounded [23]. Once $\tilde{\mathbf{a}}$ is found, the AoA estimates are the peaks in $\tilde{\mathbf{a}}$. An illustrative example is given in Fig. 3. Note that ℓ_1 based algorithms have global convergence regardless of the initialization [22], i.e., insensitive to initialization.

3.2 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 [25] 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 narrowband 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 observe that AoAs introduce no much difference across subcarriers, but ToAs contribute to measurable phase shifts across subcarriers. For example, for the kth 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

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

Fig. 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).

measure. By contrast, even if the ToA of the kth, 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,

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

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: $\{\tilde{\tau^1}, \tilde{\tau^2}, \ldots, \tau^{\tilde{N}_\tau}\}$ for To A, and $\{\tilde{\theta^1}, \tilde{\theta^2}, \ldots, \theta^{\tilde{N}_\theta}\}$ for AoA. The range of this ToA grid is $[0, \tau_{max}]$, where $\tau_{max} = 1/f_\delta$. For example, if Intel 5,300 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 5,300 cards, to jointly estimate ToAs and AoAs, we stack all the subcarrier measurements and linearized steering matrices together as follows

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

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

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.

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

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

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

3.3 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 [24], 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, ROArray is highly suitable for applications that concern

accuracy more than computation time/power, especially when low SNRs are present.

3.4 Multi-Packet Fusion

While ROArray can work well 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 [26] and use the Singular Value Decomposition (SVD) to simultaneously reduce the problem size efficiently and maintain the high performance. Nevertheless, there are two key differences in our approach. First, our scheme builds on ToA and AoA joint optimization while only AoA is concerned in [26]. Second, we need to account for the uncertain dynamic delays for different packets, which is not considered in [26] either.

Suppose we have N_t CSI measurements from a series of time samples, $\mathbf{Y} = [\mathbf{y}_{\theta\tau}^1, \ldots, \mathbf{y}_{\theta\tau}^{N_t}]$, therefore, Equation (14) becomes

$$\mathbf{Y} = \mathbf{S}_{\theta\tau}\mathbf{A} + \mathbf{N}, \tag{19}$$

where $\mathbf{A} = [\mathbf{a}_{\theta\tau}^1, \ldots, \mathbf{a}_{\theta\tau}^{N_t}]$, and $\mathbf{N}$ similarly. Solving the above equation is straightforward but time-consuming as the time cost grows superlinearly with N_t. To reduce N_t to a reasonable level, we employ SVD to separate the signal space from the noise space,[8] and derive the signal space version of Equation (19) as

$$\mathbf{Y_s} = \mathbf{S}_{\theta\tau}\mathbf{A_s} + \mathbf{N_s}, \tag{20}$$

where $\mathbf{Y_s}$ is a matrix of size $90 \times K$. $\mathbf{A_s}$ and $\mathbf{N_s}$ are similar. The core of the above processing is that $\mathbf{Y_s}$ still captures most of the signal power while the size of the time dimension is effectively reduced from N_t to K. To obtain the ToA&AoA spectrum using multiple packets, our sparse recovery problem becomes

$$\min \|\mathbf{Y_s} - \mathbf{S}_{\theta\tau}\mathbf{A_s}\|_f^2 + \kappa\|\hat{\mathbf{A}}^{\ell_2}\|_1, \tag{21}$$

$$\hat{\mathbf{A}}^{\ell_2} = [\hat{\mathbf{A}}_1^{\ell_2}, \hat{\mathbf{A}}_2^{\ell_2}, \ldots, \hat{\mathbf{A}}_K^{\ell_2}], \tag{22}$$

$$\hat{\mathbf{A}}_i^{\ell_2} = \|\hat{\mathbf{A}}_i(1), \hat{\mathbf{A}}_i(2), \ldots, \hat{\mathbf{A}}_i(K)\|_2, \tag{23}$$

where $\|\cdot\|_f$ denotes the Frobenius norm, and $\hat{\mathbf{A}}_i$ is the ith column of $\mathbf{A_s}$. The reason of using the Frobenius norm is that $\mathbf{Y_s}$ is a matrix, not a vector anymore and the optimality of using the Frobenius norm can be found at Chapter 7 of [26]. Another important thing worth noting is that the sparsity constraint of the Equation (21) is not applied on all the elements of the matrix, $\mathbf{A_s}$, but only on the ℓ_1 norm of $\hat{\mathbf{A}}^{\ell_2} = [\hat{\mathbf{A}}_1^{\ell_2}, \hat{\mathbf{A}}_2^{\ell_2}, \ldots, \hat{\mathbf{A}}_K^{\ell_2}]$, where $\hat{\mathbf{A}}_i^{\ell_2} = \|\hat{\mathbf{A}}_i(1), \hat{\mathbf{A}}_i(2), \ldots, \hat{\mathbf{A}}_i(K)\|_2$. As such, the final impact of the sparsity constraints, including κ and the ℓ_1 norm of $\hat{\mathbf{A}}^{\ell_2}$, is enforcing the sparsity only on the ToA-AoA domain but not on the packet domain, which exactly fits the real-world cases because dominant paths exhibit sparsity only in the ToA-AoA domain but exist in all the packets (and singular value basis).

One may wonder that in order to obtain the signal space, an estimate, $\hat{K}$, is still needed. We shall address this point later and show that a rough guess, which is not necessarily accurate, is enough. Another big problem is that the ground truth of ToAs across packets is hardly the same due to the dynamic packet detection delay and sample frequency offset (SFO) [2]. To solve this, we adopt the linear regression to remove the instability of ToAs across packets. For each CSI measurement of a packet, we estimate the uncertain delay as follows,

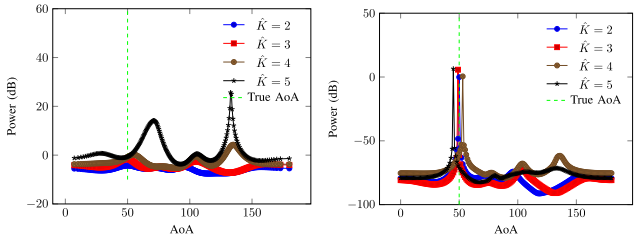
$$\hat{\tau_u} = \underset{\eta}{\operatorname{argmin}} \sum_{m,n=1}^{M,N_{sub}} \{\phi(m,n) + 2\pi f_\delta(n-1)\eta + \varepsilon\}, \tag{24}$$

where $\phi(m,n)$ is the unwrapped phase of the mth antenna at the nth subcarrier, η and ε are the slope and intercept parameters of linear regression. After obtaining the estimate, $\hat{\tau_u}$, we apply it to correct the original measurement phase as

$$\phi'(m,n) = \phi(m,n) + 2\pi f_\delta(n-1)\hat{\tau_u}. \tag{25}$$

Note that as SFO and packet detection delay are the same across antennas due to the frequency-locked loop (FLL), thus the above correction is based on all the antennas instead of any single one. Nevertheless, this correction is only good at removing the instability across packets but not making the ToA estimates close the ground truth. Thus, the resulted ToAs are still relative ToAs. As shown in Figs. 4a and 4b, the delays of paths are different for different packets. After the above correction process, solving Equation (21) finally yields sharp AoAs and stable ToAs as shown in Fig. 4c.

Although using SVD to separate the signal space from the noise space has been used in MUSIC for a long time, our way is different. MUSIC first uses SVD to find the noise space using $\hat{K}$, and then find the signal peaks that are orthogonal to it, which makes it quite sensitive to the inaccurate $\hat{K}$. In contrast, ROArray uses SVD to "compress" multiple packets and enforces sparsity on the spatial and frequency domains other than the time domain, keeping it from the bad influence of inaccurate $\hat{K}$. An example is shown in Fig. 5. From this figure, we can know that unlike MUSIC, ROArray is insensitive to $\hat{K}$, and is suitable for direct-path identification, which avoids spurious peaks. In practice, we can make 3 as an initial guess of K.



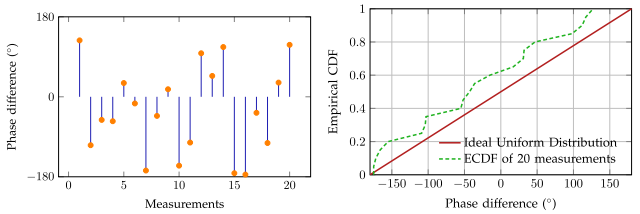
(a) SpotFi with different estimated $\hat{K}$ (b) ROArray with different estimated $\hat{K}$

Fig. 5. Comparison of SpotFi and ROArray with the sensitivity to different estimated $\hat{K}$. As expected, SpotFi is quite sensitive to different $\hat{K}$ and tends to produce spurious peaks due to the characteristics of MUSIC, whereas ROArray is insensitive to inaccurate $\hat{K}$. Note that in practice, the ground truth K is difficult to obtain, but we can use the deviation between the estimated peaks and the direct path AoA to tell which method is better.

3.5 Multi-AP Localization

At this step, ROArray attempts to localize the target by combing direct-path AoAs from several APs. Let $\{\varphi_1, \varphi_2, \ldots, \varphi_l\}$ be

8. SVD details and the relationship between the noise space and signal space are omitted here, please refer to [1], [15].



(a) Phase difference measurements for 20 times (b) ECDF of measurements in (a)

Fig. 6. The 20 times power on/off of phase difference measurements between two antennas. The results show that there are random phase offsets associated with each antenna. The distribution of these offsets is close to a uniform distribution, which is difficult to predict.

the estimated AoAs from l APs and $\{R_1, R_2, \ldots, R_l\}$ be the corresponding RSSIs. ROArray intends to localize the target by minimizing the deviation between RSSI-weighted AoAs,

$$\min \sum_{i=1}^{l} R_i(\hat{\varphi}_i - \varphi_i)^2. \tag{26}$$

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.

3.6 Phase Calibration

Every time when the working channel changes, a random phase offset will be introduced in measured CSI values. We empirically measure phase differences between 2 antennas for 20 times as an AP powers on/off. The results are shown in Fig. 6. It clearly shows the randomness of these phase offsets, which makes phase calibration necessary. Moreover, for COTS devices, offline calibration usually disrupts communication or is not easy to deploy. Therefore, we intend to design an online and multipath-compatible phase calibration scheme for COTS APs. Specifically, we employ a MUSIC-based calibration that uses the orthogonality between steering vectors and the noise space of signals. To account for random phase offsets, the Equation (13) becomes

$$s_p(\theta, \tau) = \mathbf{\Delta_p} s(\theta, \tau), \tag{27}$$

$$\mathbf{\Delta_p} = \mathrm{diag}\{\underbrace{1, \Phi_{1,1}, \ldots, \Phi_{1,M-1}}_{\text{subcarrier 1}}, \ldots, \underbrace{1, \Phi_{L,1}, \ldots, \Phi_{L,M-1}}_{\text{subcarrier L}}\}, \tag{28}$$

where $\mathbf{\Delta_p}$ is the phase offset matrix. For subcarrier i, there are $(M-1)$ unknown phase offsets to estimate, namely $\Phi_{1,1}, \ldots, \Phi_{1,M-1}$, because the first antenna is used as the basis. Therefore, in total we have $L(M-1)$ unknowns. To estimate these unknowns, we leverage two observations. First, according to MUSIC [14], we know that the steering vector should be orthogonal to the noise space of the received signals, denoted as $\mathbf{U_n}$. Second, although it is hard to get the ground truth for all the steering vectors under multipath environments, it is not a problem to measure the Line-of-Sight path, i.e., θ_L, τ_L. The above observations can be formulated as follows,

$$\|s_p(\theta_L, \tau_L)\mathbf{U_n}\|_2 = 0. \tag{29}$$

Since there are only K dominant paths (typically 5), the above equation has $LM - K$ sub-equations, which is more than the number of unknowns, $LM - L$. To further increase the number of sub-equations, we perform the above process at multiple anchor points, where the locations are measured in advance. Finally, we obtain an overdetermined system of equations and thus apply the least square method to optimize the results. The spatial smoothing is also used to combat the coherence of multipath signals [27].

Note that the involved cost of the above calibration would be negligible as it is only invoked when the AP administrator sets or changes the channel, and there is usually no channel hopping involved in regular WiFi communication. Also, the calibration process is done under high SNR scenarios to ensure the quality. While this method shares a bit common with D-Watch [28], it has several major differences. First, we correct phase offsets across both subcarriers and antennas whereas D-Watch only deals with phase offsets among antennas. Second, we need to remove the instability of ToAs across packets before calibration as in Equation (24), which is not included in D-Watch.

3.7 Other Implementation Considerations

There are a number of implementation considerations worth noting here.

Inter-Carrier Interference. Between the transmitter and receiver, both Carrier Frequency Offset (CFO) and Sampling Frequency Offset are inevitably included, resulting in inaccuracy for measured CSI values. Fortunately, the Schmidl-Cox synchronization algorithm can effectively estimate CFO, reducing the Inter-carrier Interference (ICI) to a negligible level. The ICI caused by SFO is also negligible because according to 802.11n, the SFO shall be within 20 parts per million (ppm) for the 5 GHz band and within 25 ppm for the 2.4 GHz band. Hence, the remaining SFO and CFO with off-the-shelf WiFi cards can be safely considered as noise.

Time Synchronization. For time synchronization within the antenna array, the SFO and packet detection delay does produce time shifts in our ToA estimates, which is the main reason we only use ToAs to distinguish the direct path. However, all the antennas on an AP are frequency-locked, which means the same delay being added to all the antennas [1]. Hence, they won't affect phase shifts across antennas due to AoAs and thus keep AoA estimates unchanged. Another time synchronization we need to consider is how to synchronize time among APs. For this, we put the customized packet sequence number into the payload for each packet at the transmitter and check the number at the receiver. By matching sequence numbers, it is easy to synchronize behaviors among distributed APs.

Grid Refinement. In theory, one wants the grid in $\mathbf{S}$ to be as fine as possible. However, an overly fine grid would affect the recovery accuracy due to the increasing coherence inside $\mathbf{S}$ [20]. We address this by using a gradient-based optimization approach [29] to measure the mutual coherence in $\mathbf{S}$ and thus find the optimal grid spacing.

Near-Field Signal. Although we assume far-field signals for simplicity in the previous sections, the framework of ROArray is easy to include the dimension of range. In order to localize the target in the near-field of the array, we can further parameterize the steering matrix on the dimension of range based on a grid, $\{\tilde{r^1}, \tilde{r^2}, \ldots, \tilde{r^{N_r}}\}$. Then by stacking all the vectors into a vector, we shall get the same sparse recovery form as for the far-field case. Also, sparsity should be enforced on the dimension of range due to the distinct range associated with each

independent path. The new problem can be solved similarly as Equation (18). Obviously, the MUSIC framework is hard to deal with near-field signals and does not have such convenience.

Note that our smallest ToAs for direct path identification is inspired by SpotFi [1]. Same as [1], the resolution of ToAs derived above are on the order of several nanoseconds. Thus, they are not accurate enough for small distances and sub-meter single AP localization. Actually, the main purpose of adding ToA to AoA estimation is to make AoA estimates more accurate. For example, if there are two paths (3.2 ns, 70°), (8.4 ns, 68°), such two paths are indiscernible in previous AoA methods that do not consider ToAs [12] and thus the energy of two paths are merged into one. Nevertheless, for our method and SpotFi, both two paths are easy to separate, making AoA estimates more accurate. But sometimes the path with the smallest ToA may not be the direct path when the direct path is too weak. Hence, all the ToA/AoA estimates would be for reflected paths. To deal with such cases, our solution, which uses multi-packet fusion, can increase the discernability of weak signals by reinforcing the signal space through a series of time samples. This idea appears similar to SpotFi as well. But SpotFi uses a non-coherent approach, clustering, while our multi-packet fusion is a coherent approach ensuring that the found direct path exists on the same signal subspace of the time domain.

Array Geometry. While the principle of ROArray can be generalized to more flexible array geometry, there are several issues worth further investigation. The first is how to choose the best (irregular) array placement. Usually, having array elements spaced further away makes beams sharper but comes with more ambiguity while reducing the array spacing can remove the ambiguity but also brings down the resolution. Hence, finding the best array placement is the key to optimizing ROArray for arbitrary array geometry. This issue becomes even more difficult when someone wants to use channel hopping to increase the resolution of ToA, because WiFi bands are not evenly distributed across 2.4/5 GHz. While this paper mainly focuses on the solution and validation of ROArray under regular spacing at half of the wavelength, similar to [1], [12], we leave optimizations of ROArray for irregular arrays for future work.

4 Experiments

4.1 Implementation

To verify the design of ROArray, we implement it using off-the-shelf Intel 5,300 WiFi cards. Linux CSI Tools [30] 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 5,300 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 5,300 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. The CSI tools [30] can output an SNR for each subcarrier. We use the average of all subcarriers to denote the SNR of an antenna.

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 [31] 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 Fig. 8. In total, we tested 300 different locations. Note that the sliding doors and chairs are mainly used to test for NLoS scenarios.

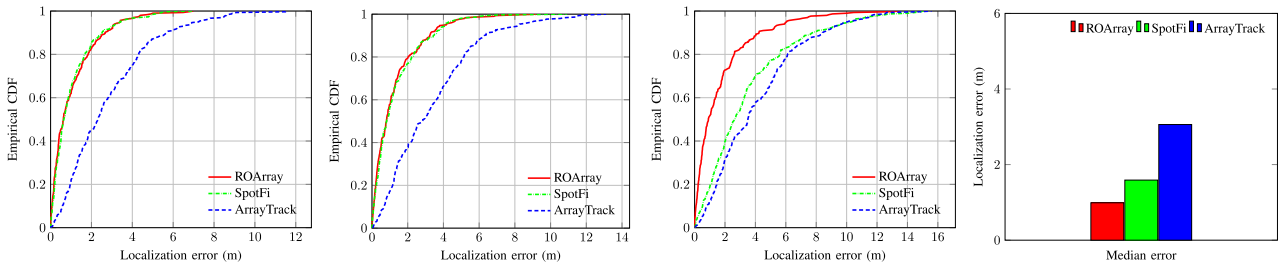
We compare ROArray with state-of-the-art AoA based WiFi systems, SpotFi [1] and ArrayTrack [12]. WiDeo [32] is not included, because it requires software defined radios to implicitly eliminate packet detection delay (and the fractional SFO), which is not compatible with Intel 5,300 cards. Ubicarse [8] and CUPID [6] are not 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 fair competition. For phase calibration, we compare ours with Phaser [13]. But Phaser's localization scheme is not included due to its additional hardware requirements, such as antenna rerouting,

Doing experiments with different SNRs usually has two ways. The one way is to use the Transmit Power Control mechanism, which is specified in WiFi standards. Nevertheless, not all commercial NICs support this function very well. For example, in order to obtain CSI, we choose to use Intel 5,300 WiFi cards, whose power is fixed 15 dBm and does not allow any changes. The other solution is to group the results based on SNRs across various locations, which applies to all commercial WiFi cards. So we adopt the second solution in our experiments.

4.2 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 Fig. 7.

We observe from Fig. 7a 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 90th 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



(a) High SNRs, ≥15 dB (b) Medium SNRs, (2,15) dB (c) Low SNRs, ≤2 dB (d) Overall comparison for all SNRs

Fig. 7. Comparison of ROArray's localization accuracy with SpotFi and ArrayTrack under high, medium, and low SNR scenarios.

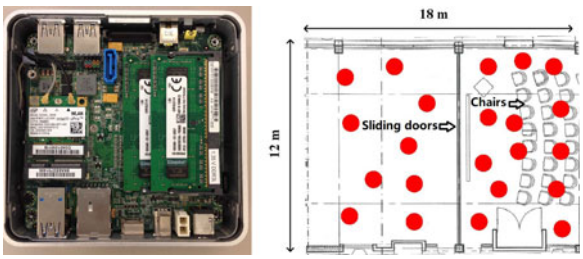
medium SNRs in Fig. 7b. As expected, when SNRs decrease, the performances of all the three systems deteriorate accordingly. However, when SNRs drops to a low level as shown in Fig. 7c, the median accuracies of SpotFi and ArrayTrack degrade to 2.61 m and 3.52 m respectively whereas ROArray achieves 0.91 m. Such performance gain of ROArray over other systems largely comes from the robustness of sparse recovery techniques. While ArrayTrack appears to be not that affected by different SNRs, its median accuracy actually degrades from 2.32 m for high SNRs and 2.81 m for medium SNRs to 3.5 m for low SNRs. The main reason that ArrayTrack looks like 'robust' to all SNRs is that its localization accuracy at high SNRs is very low compared to SpotFi and ROArray due to its limited resolution.

Also, the overall performance comparison of three systems under all SNRs is shown in Fig. 7d. In particular, ROArray achieves 0.99 m median accuracy while SpotFi and ArrayTrack have 1.59 m and 3.06 m median errors. This again shows ROArray is better than SpotFi and ArrayTrack when there are different SNR cases for indoor environments.

4.3 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 evaluation with three different SNR situations. Fig. 9a 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 Fig. 9b, 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 of SpotFi is its sensitivity to inaccurate $\hat{K}$.[9]



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

Fig. 8. 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.

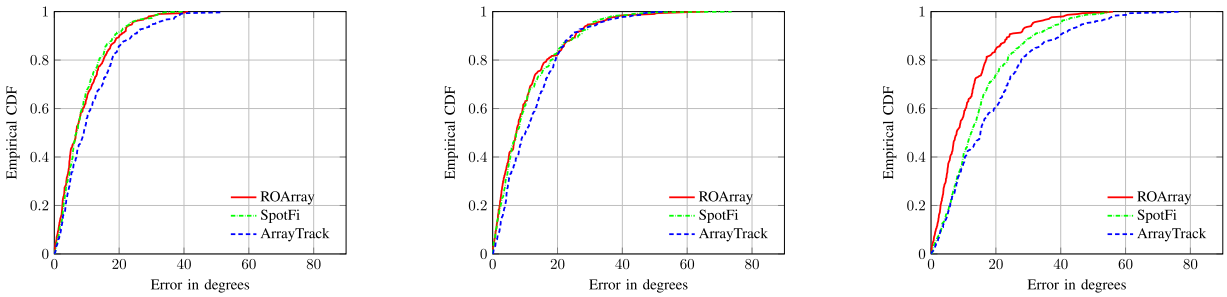
4.4 Impact of Phase Calibration

Next, we examine the performance of our phase calibration method and compare it to Phaser [13] in detail. The results are plotted in Fig. 10. First, we compare it with Phaser regarding absolute calibration error. As shown in Fig. 10a, as the number of anchor points increases, the performance gain of ROArray is much better than Phaser. Specifically, When the number of anchor point is 6, Phaser's calibration error is 30.2 degree while ROArray's error is only 12.1 degree. This is mainly because Phaser is not robust to multipath environments, unlike ROArray where we only rely on LoS paths. Another contributor may come from the joint estimation across subcarriers while Phaser treats each subcarrier independently. Due to such performance gain, it is not surprising to see that the AoA estimation of ROArray is also much better than that of Phaser. In particular, the median AoA error of ROArray is 9.8 degree whereas its of Phaser is 20.2 degree, as shown in Fig. 10b. The localization accuracy comparison using two different phase calibration method is shown in Fig. 10c. ROArray's calibration achieves an improvement of 1.38 m over Phaser in terms of median localization accuracy. This huge performance gain stems from the MUSIC-based calibration scheme and least square optimization using multiple anchor points.

4.5 Impact of Multi-Packet Fusion

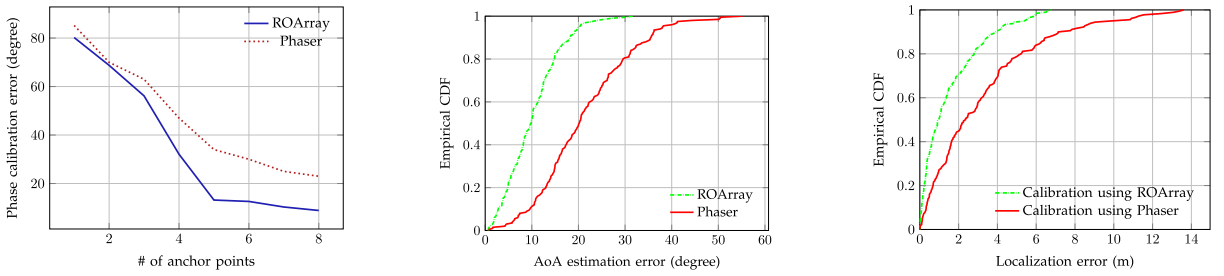
To investigate the impact of multi-packet fusion, we conduct experiments under different multipath and SNR scenarios.

9. 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

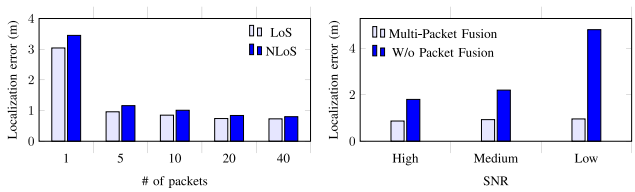
Fig. 9. Comparison of ROArray's AoA estimation errors with SpotFi and ArrayTrack under high, medium, and low SNR scenarios.



(a) Absolute calibration error (b) AoA comparison (c) Localization accuracy comparison

Fig. 10. Comparison of ROArray and Phaser in terms of absolute calibration error, AoA estimation error, and localization error. Those results show that ROArray significantly outperforms Phaser in phase calibration.

First, we examine the impact of packet numbers. We vary the number of sampled packets from 1 to 40, and group results into two categories, LoS and NLoS. As shown in Fig. 11a, the accuracy improves as more packets are collected for both LoS and NLoS cases. In particular, the average localization error is 3.04 m for LoS cases when only 1 packet is involved. It improves to 0.74 m when 20 packets are used. At least two factors are contributing to this. First, ROArray achieves better accuracy by coherently combing multiple samples using SVD decomposition. Second, multiple samples intuitively help to filter out unpleasant outliers. Note that the gain of more packets is diminishing and almost negligible when the packet number reaches 40. To better see the impact of multi-packet fusion, we further compare it with the scheme without fusion. As shown in Fig. 11b, unsurprisingly the scheme without multi-packet fusion is much worse than the scheme with fusion. We observe that especially in low SNR cases, the median accuracy of the scheme without fusion degrades from 1.8 m with high SNRs to 4.8 m with low SNRs. Meanwhile, the accuracy of ROArray keeps stable under 1 m, which shows the power of multi-packet fusion that coherently compresses multiple packets.

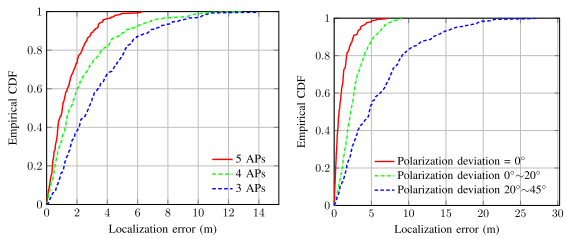


(a) Localization errors with different # of packets (b) Localization errors with and without multi-packet fusion

Fig. 11. The impact of multi-packet fusion of ROArray in both LoS and NLoS cases and various SNR cases. ROArray shows significant improvement on robustness thanks to the multi-packet fusion.

4.6 Varying Number of APs

Next, we investigate how AP density impacts ROArray's accuracy. By varying the number of APs that can hear the client from 3 to 5, we present results in Fig. 12a. 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.



(a) Localization errors with different # of APs (b) Localization errors with different polarization deviation angles

Fig. 12. (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 deviation angles of antenna polarization, indicating the importance of polarization.

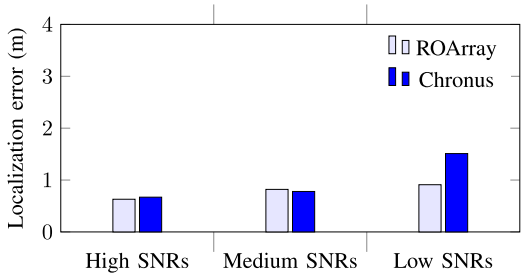


Fig. 13. Localization errors under different SNRs for ROArray and Chronus.

4.7 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 Fig. 12b 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 $0°\sim 20°$ and $20°\sim 45°$ 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.

4.8 ToA-Based Method Comparison

We also compare ROArray with Chronus [2], a state-of-the-art ToA-based solution using off-the-shelf devices. The comparison criterion is based on different levels of SNRs. For Chronus, we implement it on all the available bands for both 2.4&5 GHz and obtain a CSI for each band while hopping channels. The results in Fig. 13 demonstrate that ROArray'accuracy is comparable with Chronus when SNRs are high and medium. Also, Chronus only achieves 1.51 m median accuracy for low SNRs while ROArray maintains sub-meter localization errors. This is mainly because Chronus does not have mechanisms that can deal with low SNRs. Note that fair competition for AoA and ToA systems is really difficult as there are too many different impacting factors. For example, Chronus uses only a single-AP but ROArray requires multiple APs. ROArray is fully compatible with ongoing WiFi communication while Chronus has to disrupt communication due to channel hopping. Actually, both systems have their own suitable applications. Chronus' single-AP solution is fit for small businesses that are concerned about device cost While ROArray is more suitable for large-scale deployment where requires stable and communication-compatible localization services.

5 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 [33], [34]. 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 [32], 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 in 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 away 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 [34]. 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.

Others. Many sensor fusion based systems are also proposed to further improve the accuracy of pure WiFi based methods [6], [7], [8], [35]. They usually make use of sensors, such as gyroscopes, accelerometers, to derive walking distances, orientations and moving directions. Together with floor plans, those deduced mobile metrics can effectively

improve the accuracy of smartphone users. Meanwhile, a great many RFIDs based localization systems also achieve impressive performance with different application scenarios [36], [37]. However, we think they are not as ubiquitous as widely deployed WiFi infrastructure around the world.

Sparse recovery based AoA estimation is extensively studied in signal processing and information theory areas [26], [38], [39], 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.

6 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.

As the current time-cost of our system is still high and unfit for real-time tracking demands, there are several possible ways for further exploration. The first way could be to introduce fast single-packet processing techniques, such Matrix Pencil [40]. Such integration requires a coherent detector because random phases can make the average tend to zero [40]. The second potential optimization is to try Sparse Bayesian Inference that can maintain high estimation accuracy even under a very coarse sampling grid. One of the issues with this is that how to adaptively adjust grid spacing according to different SNRs [38]. The third way is to use wide-band WiFi, e.g., 802.11 ad with 60 GHz, that can provide ToA with much higher resolution. Nevertheless, such methods are hard to implement on off-the-shelf devices and are barely able to handle NLoS cases.

Acknowledgments

This work was supported in part by a Canada Technology Demonstration Program Grant, a Canada NSERC Discovery Grant, an NSERC E.W.R. Steacie Memorial Fellowship, and NSFC under Grant 61472268.

References

[1] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. of ACM Conf. Special Interest Group Data Commun.*, 2015, pp. 269–282.

[2] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. 13th Usenix Conf. Netw. Syst. Des. Implementation*, 2016, pp. 165–178.

[3] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

[4] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: Locating devices in an office environment," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 281–292.

[5] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.

[6] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2013, pp. 249–262.

[7] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "SAIL: Single access point-based indoor localization," in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 315–328.

[8] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 483–494.

[9] Wi-Fi Tags, (2018). [Online]. Available: 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. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 53–64.

[11] J. Xiong, K. Sundaresan, and K. Jamieson, "ToneTrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 537–549.

[12] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. 10th USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 71–84.

[13] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity WiFi access points," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 153–164.

[14] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[15] P. Stoica and N. Arye, "MUSIC, maximum likelihood, and cramer-rao bound," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 5, pp. 720–741, May 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. 10th ACM Int. Conf. Emerging Netw. Experiments Technol.*, 2014, pp. 41–52.

[17] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical many-antenna base stations," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 53–64.

[18] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Resonance Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[20] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2182–2195, May 2011.

[21] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*, vol. 1. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[22] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization," *Proc. Nat. Acad. Sci. United States America*, vol. 100, no. 5, pp. 2197–2202, 2003.

[23] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, no. 1, pp. 193–228, 1998.

[24] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[25] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li , "LTE radio analytics made easy and accessible," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 211–222.

[26] D. Malioutov, "A sparse signal reconstruction perspective for source localization with sensor arrays," Master Thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, 2003.

[27] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 33, no. 4, pp. 806–811, Aug. 1985.

[28] Y. Wang, J. Xiong, H. Jiang, X. Chen, and D. Fang, "D-Watch: Embracing bad multipaths for device-free localization with COTS RFID devices," in *Proc. 12th Int. Conf. Emerging Netw. EXperiments Technol.*, 2016, pp. 253–266.

[29] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei, "On optimization of the measurement matrix for compressive sensing," in *Proc. 18th Eur. Signal Process. Conf.*, 2010, pp. 427–431.
[30] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, 2011.
[31] CVX solvers, (2012). [Online]. Available: http://cvxr.com/cvx/doc/solver.html
[32] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "WiDeo: Fine-grained device-free motion tracing using RF backscatter," in *Proc. 12th USENIX Conf. Netw. Syst. Des. Implementation*, 2015, pp. 189–204.
[33] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surv. Tuts.*, vol. 11, no. 3, pp. 107–124, Jul.–Sep. 2009.
[34] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, 2013, Art. no. 25.
[35] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of WiFi based localization for smartphones," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 305–316.
[36] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SGICOMM*, 2015, pp. 235–246.
[37] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 237–248.
[38] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse Bayesian inference," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 38–43, Jan. 2013.
[39] M. M. Hyder and K. Mahata, "Direction-of-arrival estimation using a mixed l2,0 norm approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4646–4655, Sep. 2010.
[40] Y. Hua and T. K. Sarkar, "Method for estimating parameters of bxponentially damped/undamped sinusoids in noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 5, pp. 814–824, May 1990.

Wei Gong (M'14) After receiving the BE degree in computer science from the Huazhong University of Science and Technology, he went to Tsinghua University, where he received the ME degree in software engineering and the PhD degree in computer science. He is a professor in the School of Computer Science and Technology, University of Science and Technology of China. His research focuses on wireless networks, Internet-of-Things, and distributed computing. He had also conducted research at Simon Fraser University and the University of Ottawa. He is a member of the IEEE.

Jiangchuan Liu (S'01-M'03-SM'08-F'17) received the BEng. degree (cum laude) in computer science from Tsinghua University, Beijing, China, in 1999, and the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2003. He is currently a university professor with the School of Computing Science, Simon Fraser University, BC, Canada. He is an NSERC E.W.R. Steacie Memorial Fellow. He is a steering committee member of the *IEEE Transactions on Mobile Computing*. He was a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM in 2015, the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award in 2013, and the ACM Multimedia Best Paper Award in 2012. He has served on the editorial boards of the *IEEE/ACM Transactions on Networking*, the *IEEE Transactions on Big Data*, the *IEEE Transactions on Multimedia*, the *IEEE Communications Surveys and Tutorials*, and the *IEEE Internet of Things Journal*. He is a fellow of the IEEE.